



저작자표시-비영리-변경금지 2.0 대한민국

이용자는 아래의 조건을 따르는 경우에 한하여 자유롭게

- 이 저작물을 복제, 배포, 전송, 전시, 공연 및 방송할 수 있습니다.

다음과 같은 조건을 따라야 합니다:



저작자표시. 귀하는 원저작자를 표시하여야 합니다.



비영리. 귀하는 이 저작물을 영리 목적으로 이용할 수 없습니다.



변경금지. 귀하는 이 저작물을 개작, 변형 또는 가공할 수 없습니다.

- 귀하는, 이 저작물의 재이용이나 배포의 경우, 이 저작물에 적용된 이용허락조건을 명확하게 나타내어야 합니다.
- 저작권자로부터 별도의 허가를 받으면 이러한 조건들은 적용되지 않습니다.

저작권법에 따른 이용자의 권리는 위의 내용에 의하여 영향을 받지 않습니다.

이것은 [이용허락규약\(Legal Code\)](#)을 이해하기 쉽게 요약한 것입니다.

[Disclaimer](#)

생활과학석사 학위논문

스크립트 기반의 범용
3차원 인체 측정 시스템 개발

2018년 8월

서울대학교 대학원

의류학과 패션테크놀로지 전공

김 민 석

스크립트 기반의 범용 3차원 인체 측정 시스템 개발

지도교수 김 성 민

이 논문을 생활과학석사 학위논문으로 제출함
2018년 4월

서울대학교 대학원
의류학과 패션테크놀로지 전공
김 민 석

김민석의 생활과학석사 학위논문을 인준함
2018년 6월

위 원 장 남 윤 자 (인)

부위원장 송 화 경 (인)

위 원 김 성 민 (인)

국문초록

본 연구에서는 인체 스캐너 종류와 관계없이 3차원 인체 스캔 데이터로부터 랜드마크를 검색하고 각종 치수를 측정하는데 필요한 3차원 인체 측정 시스템을 개발하였다. 이를 위해 여러 인체 스캐너에서 생성된 다양한 형식의 3차원 형상 데이터를 입출력에 필요한 중립적 데이터 구조로 설계하고, 설계한 데이터 구조로 변환할 수 있는 데이터 변환 소프트웨어를 개발하였다. 또한, 측정해야 할 랜드마크나 치수를 소프트웨어에 미리 정의해 두지 않고 시스템에서 개발된 스크립트 언어로 사용자가 측정하기 원하는 랜드마크나 치수를 직접 정의할 수 있게 소프트웨어 개발하여 프로그래밍에 대한 지식이 없는 사용자라도 인체 스캔 데이터의 특성에 따라 자유롭게 인체 측정을 실행할 수 있도록 하였다. 이뿐만 아니라 살점이나 겨드랑점 같이 인체 측정에 중요하지만 정의하기 까다로운 랜드마크를 자동으로 찾는 함수도 추가하였으며, 직접 만지거나 뼈를 통해 검색할 수 있는 랜드마크를 측정하기 위해 인공지능망을 활용하여 자동으로 랜드마크를 검색하는 기능도 개발하였다. 끝으로, 측정 결과를 다양한 방법으로 시각화하여 데이터 분석 작업이 용이하도록 개발하였으며, 이와 같은 모든 작업이 편리하도록 통합 개발 환경으로 3차원 인체 측정 시스템을 개발하였다.

주요어 : 3차원 인체 스캔 데이터, 인체 측정, 범용 소프트웨어, 스크립트 기반 측정, 랜드마크 검색, 통합 개발 환경

학 번 : 2016-25278

목 차

제 1 장 서 론	1
제 1 절 연구의 필요성	1
제 2 절 연구의 목적	4
제 2 장 선행 연구	5
제 1 절 자동 랜드마크 측정	5
제 2 절 자동 인체 측정 상용 소프트웨어	8
제 3 장 연구 방법	10
제 1 절 인체 모델 데이터 설계	10
제 2 절 인체 측정 소프트웨어 설계	12
제 4 장 연구 결과 및 고찰	14
제 1 절 데이터 변환 소프트웨어	14
제 2 절 통합 개발 환경	15
제 3 절 스크립트 시스템과 명령어 정의	17
1. 기본 명령어	19
2. 기하학 관련 명령어	20
3. 측정 관련 명령어	27

4. 특수 명령어	34
제 4 절 스크립트 작성	42
1. 인체 영역 분할	42
2. 영역 분할 활용	45
제 5 절 기계 학습을 활용한 랜드마크 검색	48
제 5 장 결론 및 제언	51
참고문헌	53
Abstract	56
부 록	58

표 목 차

[표 1] 기계 학습 데이터	38
[표 2] 인공신경망 구성	40
[표 3] 인체 영역 분할을 위한 스크립트	43
[표 4] 허리 관련 랜드마크 검색을 위한 스크립트	46

그 립 목 차

[그림 1] 3차원 모델의 데이터 구조	10
[그림 2] 인체 측정 소프트웨어 구조	13
[그림 3] 데이터 변환 소프트웨어	14
[그림 4] 통합 개발 환경	15
[그림 5] 명령어 종류	18
[그림 6] 영역 정의를 위한 도식화	22
[그림 7] UnionRegion의 예	23
[그림 8] 단면 분석의 예	26
[그림 9] 측정 결과 예시	28
[그림 10] 직선 측정 시각화 예시	29
[그림 11] 두 점 사이의 패스 정의 사례	31
[그림 12] 각도 측정 사례	32
[그림 13] 부피 계산을 위한 볼륨 메쉬 생성 사례	33
[그림 14] 인체 실루엣 분석 도식화	35
[그림 15] 각 실루엣 점으로부터 원점까지의 거리 측정 ·	36
[그림 16] 인체신경망 학습 사례	39
[그림 17] 인체신경망을 활용한 랜드마크 측정 결과	41
[그림 18] 인체 영역 분할 결과	44
[그림 19] 인체 모델에서 얻은 허리 단면	47
[그림 20] 개발된 3차원 인체 측정 시스템의 개요	48
[그림 21] 다양한 인체 모델 측정 결과	49

제 1 장 서 론

제 1 절 연구의 필요성

최근 패션, 헬스케어, 엔터테인먼트 등의 여러 산업 분야에서 3차원 인체 데이터의 활용이 증가하고 있다 (Treleaven and Wells, 2007, Werghi, 2007). 특히, 패션산업 분야에서는 치수 조사, 의류생산용 마네킹 제작, 개별 맞춤형 제작, 온라인상에서의 가상 착의 등에 3차원 인체 데이터가 활용되고 있으며 (D'Apuzzo, 2007), 학문적으로는 새로운 치수 체계 제안, 평면 전개의 따른 패턴 제작, 효율적 그레이딩 방법 제안, 가상현실 패션 디자인 연구 개발 등에 활용되고 있다 (Petrova and Ashdown, 2008). 이와 같이 다양한 분야에서의 3차원 인체 데이터 활용이 결실을 맺기 위해서는 정확한 인체 측정이 선행되어야 한다. 왜냐하면, 인체 측정으로 얻은 데이터가 대부분의 기술 및 연구의 기초 자료로 사용되기 때문이다.

3차원 인체 스캐너로부터 얻은 데이터를 분석하여 인체를 측정하는 과정은 크게 측정의 기준점이 되는 랜드마크를 찾는 작업과 랜드마크 사이에 측정 경로를 만들어 치수를 측정하는 작업으로 나눌 수 있으며, 이 과정에서 랜드마크를 찾는 작업이 치수를 측정하는 작업보다 더 어렵다 (Han and Nam, 2011). 그 이유는 비접촉 방식으로 스캔하는 3차원 인체 스캐너의 특성상 만지거나 뼈를 통해 검색할 수 있는 랜드마크를 측정하기 어렵기 때문이다 (Devarajan and Istook, 2004). 따라서 인체 데이터 분석에 관한 연구도 주로 랜드마크 검색에 중점을

두고 진행되었으며 3차원 인체 스캔 데이터로부터 자동으로 랜드마크를 추출하기 위해 오랜 기간에 걸쳐 많은 연구가 이루어졌으나(Dekker et al., 1999, Suikerbuik et al., 2004, Azouz et al., 2006, Giachetti et al., 2014), 체형에 구애받지 않고 안정적으로 랜드마크를 찾을 수 있는 방법이 아직 개발되지 않았다.

한편, 여러 회사에서 다양한 측정 방식의 3차원 인체 스캐너를 상용화하였으며 대부분 인체 측정 소프트웨어도 함께 제공하고 있다. 그러나 한 회사의 소프트웨어를 다른 회사의 스캐너에서는 사용할 수 없다는 문제가 있다 (D'Apuzzo, 2007). 또한, 사전에 정의된 랜드마크나 측정치 이외에 새로운 랜드마크나 측정 항목을 정의하여 측정할 수도 없다 (Simmons and Istook, 2003).

따라서 본 연구에서는 위와 같은 문제점을 해결하기 위해 새로운 인체 측정 시스템을 개발하고자 하였다. 먼저, 기존의 인체 측정 소프트웨어는 특정 인체 스캐너에서 얻은 특정 형식의 데이터만 분석 및 측정이 가능하다. 따라서 스캐너 종류와 관계없이 여러 스캐너에서 얻은 다양한 형식의 스캔 데이터를 새로 개발한 인체 측정 소프트웨어에서 사용할 수 있도록 데이터를 변환하는 프로그램을 개발하였다. 또한, 기존의 인체 측정 소프트웨어는 새로운 랜드마크나 측정 항목을 정의할 수 없다. 따라서 사용자가 다른 랜드마크나 측정 항목을 얻기 위해서는 수동으로 일일이 작업하거나 프로그램을 직접 개발하여야 했다. 이를 해결하기 위해 프로그래밍 언어보다 훨씬 간단한 스크립트 언어를 개발하였다. 개발한 스크립트 언어를 이용하여 사용자가 직접 랜드마크나 측정치를 정의할 수 있으며, 미리 작성한 스크립트를 활용하여 많은 양의 인체 데이터를 쉽게 측정할 수 있는 소프트웨어를 개발하였다. 또한, 인체 측정에 특화된 여러 함수 및 기능을 개발하여 효과적인 측정이 이루어질 수 있도록

다양한 명령어를 제공하였다. 마지막으로 이와 같은 작업을 통합 개발 환경(Integrated Development Environment)에서 수행할 수 있도록 인체 측정 시스템을 개발하였다.

제 2 절 연구의 목적

본 연구의 목적은 다양한 종류의 3차원 인체 스캐너로부터 얻은 데이터를 이용하여 사용자가 원하는 랜드마크와 치수를 자유롭게 정의하고 측정할 수 있는 인체 측정 시스템을 개발하는 것이며, 구체적 목적은 다음과 같다.

첫째, 스캐너 종류와 관계없이 3차원 인체 데이터를 분석할 수 있는 인체 측정 시스템을 개발하고자 하였다. 이를 위해 중립적 데이터 구조를 새롭게 설계하였다. 그리고 인체 측정 소프트웨어에서 사용할 수 있도록 다양한 형식의 스캔 데이터를 설계한 데이터 구조로 변환하는 프로그램을 개발하였다.

둘째, 사용자가 직접 측정하고자 하는 랜드마크와 치수를 쉽게 정의하고 측정할 수 있는 인체 측정 시스템을 개발하고자 하였다. 이를 위해 복잡하고 어려운 프로그래밍 언어보다 훨씬 간단한 스크립트 언어를 개발하였다.

끝으로 복잡한 3차원 인체 측정을 비교적 쉽게 수행할 수 있는 인체 측정 시스템을 개발하고자 하였다. 이를 위해 인체 측정의 전 과정을 대화식 사용자 인터페이스로 수행할 수 있도록 통합 개발 환경으로 인체 측정 시스템을 개발하였다.

제 2 장 선행 연구

제 1 절 자동 랜드마크 측정

이미 언급했듯이 3차원 인체 데이터 분석 과정에서 치수를 측정하는 작업보다는 랜드마크를 검색하는 작업이 더 어렵다. 따라서 인체 데이터 분석에 관한 연구 역시 자동 랜드마크 측정 연구 위주로 진행되었다.

Dekker et al. (1999)은 3차원 인체 모델을 기하학적으로 분석하고 유용한 연산자로 라이브러리를 구축하였으며, 재사용이 가능한 연산자의 조합을 이용하여 랜드마크를 자동으로 찾고자 하였다. 그러나 상대적으로 찾기 쉬운 일부 랜드마크만을 자동으로 추출하였으며, 찾기 어려운 랜드마크는 수동으로 표시하는 방법을 사용하였기 때문에 완전한 자동 랜드마크 추출 방법이라 할 수 없다. 또한, 항상 3차원 인체 스캔 데이터가 올바르게 정렬되어 있다고 가정하고 연구를 진행하였으나, 스캐너의 종류에 따라 인체 데이터 형식이 다르기 때문에 3차원 인체 스캔 데이터가 항상 올바르게 정렬되어 있는 것은 아니다. 따라서 인체 스캔 데이터 정렬을 위한 명령어가 추가로 필요하다. 또한, BMI 지수가 15 ~ 25 사이의 인체 데이터에서만 랜드마크 검색이 잘 작동하도록 알고리즘을 구성하였기 때문에 BMI 지수가 15 ~ 25 사이의 범위를 벗어나는 인체 데이터에서는 랜드마크 검색이 잘 작동하지 않는 한계가 있다.

Suikerbuik et al. (2004)은 미국과 유럽 시민들의 인체 정보 데이터베이스인 CAESAR(Civilian American and European Surface

Anthropometry Resource) 중 네덜란드 데이터베이스를 이용하여 모든 인체 스캔 데이터의 랜드마크가 존재하는 범위를 한정하고, 이 범위 안에서 템플릿 매칭, 함수 피팅, 곡률 계산 방법으로 랜드마크를 검색하고 측정 결과를 비교하였다. 템플릿 매칭은 모든 랜드마크 정보가 담겨있는 하나의 템플릿과 비교하여 랜드마크를 추출하는 방법이고, 함수 피팅은 랜드마크 주변을 모사할 수 있는 적당한 함수를 통해 랜드마크를 추출하는 방법이며, 곡률 계산은 관심 영역에서의 곡률을 계산하여 랜드마크를 추출하는 방법이다. 템플릿 매칭이 정확도가 가장 높은 방법이나 측정 시간이 너무 오래 걸리는 단점이 있으며, 정확도가 조금 떨어지더라도 측정 시간을 줄이기 위해 함수 피팅, 곡률 계산 방법을 개발하고 비교하였다. 그러나 오직 5가지 랜드마크만을 대상으로 실험을 진행했기 때문에 다른 랜드마크들에 대해서도 검증이 필요하며, 네덜란드인의 데이터베이스만을 사용하여 랜드마크를 검색하였기 때문에 네덜란드인과 체형 특징이 다른 인종에서는 랜드마크를 찾기 어려운 한계가 있다.

Azouz et al. (2006)은 CAESAR의 랜드마크 데이터를 페어와이즈 마르코브 네트워크(Pairwise Markov network)으로 학습하여 최적의 랜드마크 세트를 찾아 템플릿 모델을 만들고, 이 템플릿 모델을 통해 자동으로 랜드마크를 찾는 연구를 수행하였다. 200개의 인체 스캔 데이터로 학습시키고 30개의 인체 스캔 데이터로 이를 검증하였으며 73개의 랜드마크를 자동 추출하였다. 검증 결과 대부분의 랜드마크에서 평균 2cm 미만의 오차가 발생하였으나 이는 수동으로 랜드마크를 찾는 전통적인 방법의 허용 오차 1cm보다 큰 수준이었으며, 가장 큰 오차는 6.2cm이었다. 이와 같이 랜드마크의 오차가 큰 이유는 인체 스캔 데이터의 다양성에서 기인한다고 볼 수 있다.

2014년에는 수동으로 표시한 랜드마크의 위치를 인식하여 랜드마크를 자동으로 측정하는 경연대회 SHREC 2014가 개최되었다 (Giachetti et al., 2014). 각 참가자에게 수동으로 6개의 랜드마크가 표시된 인체 모델 50세트를 훈련용으로 제공하고, 랜드마크가 표시되지 않은 다른 인체 모델 50세트를 테스트용으로 제공하였다. 테스트는 각 랜드마크마다 참가자가 자동으로 찾은 랜드마크와 수동으로 표시된 랜드마크의 거리를 측정하는 방식으로 이루어졌다. 자동 랜드마크 측정을 위해 그래픽 모델법, 표면 대 표면 등록법 등의 다양한 방식으로 6팀이 참가하였으나, 가장 우수한 팀조차 6개의 랜드마크에서 평균 21.44 mm의 오차가 발생하였다. 이 경연대회 결과는 다양한 체형에 대해 자동으로 랜드마크를 추출하는 것이 어려움을 보여주는 좋은 사례이다. 랜드마크 추출의 정확도를 높이기 위해서는 여러 제약 조건을 추가해야 한다. 따라서 제약 조건을 추가하면 추가할수록 랜드마크 추출의 정교함이 높아진다. 그러나 제약 조건이 많아지면 많아질수록 다양한 인체 형상이나 크기에 따라 제약 조건을 벗어나는 특이점이 발생하고 이로 인해 오차가 크게 발생하기 때문에 다양한 체형에 대해 높은 정확도로 랜드마크를 자동 측정하기란 매우 어려운 과제이다.

제 2 절 자동 인체 측정 상용 소프트웨어

서론에서도 이미 언급했듯이 3차원 인체 스캐너를 개발하고 상용화한 회사에서는 스캐너에서 얻은 데이터를 시각화하고 분석 가능하며 측정할 수 있는 3차원 자동 인체 측정 소프트웨어도 함께 제공하고 있다. 이러한 대표적인 인체 측정 소프트웨어로는 TC²(USA)의 Body Measurement System(BMS), Telmat(FRANCE)의 SYMCAD, Hamamatsu(JAPAN)의 BL Manager, Wicks&Wilson(UK)의 TriForm, Cyberware(USA)의 DigiSize 등이 있다 (Istook and Hwang, 2001). 그러나 각각의 인체 측정 소프트웨어가 스캔 과정에서 발생하는 데이터 유형에 맞게 특별히 설계되었기 때문에 스캔 데이터 형식과 관계없이 범용으로 사용할 수 있는 인체 측정 소프트웨어가 존재하지 않는다 (D'Apuzzo, 2007).

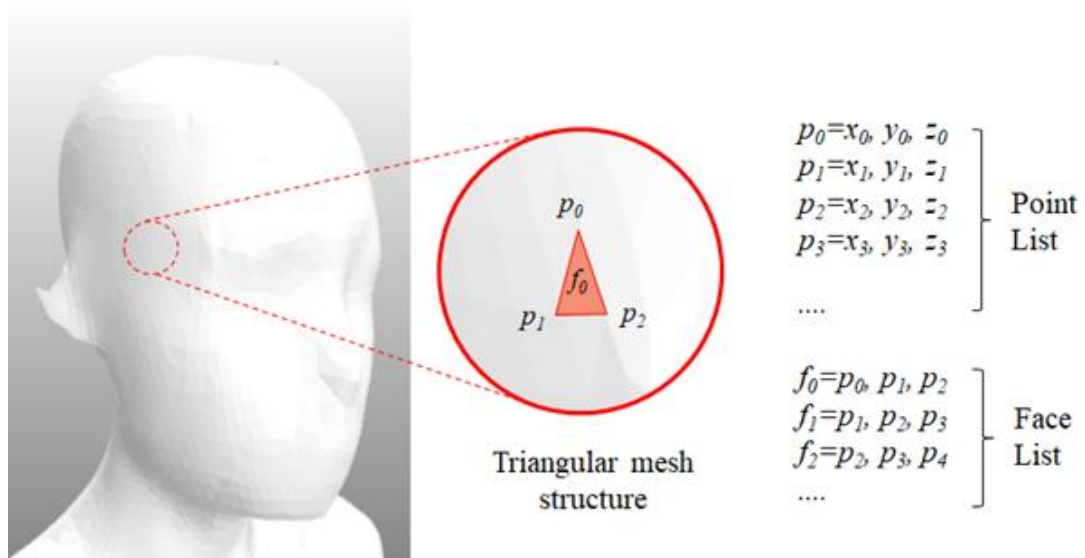
Simons and Istook (2003)은 3차원 인체 스캔 기술을 이용한 인체 측정으로 얻은 인체 측정 데이터를 의류생산에 적용하고자 하였다. 의류를 만드는데 필요한 필수 인체 측정 치수 21개를 정의하고 이를 찾기 위해 3가지 스캐너의 자동 인체 측정 상용 소프트웨어(BMS, DigiSize, SYMCAD)을 사용하였으며, 측정 결과를 비교하고 분석하고자 하였다. 실험 결과, 스캐너마다 21개 측정 항목 중 일부 측정되지 않는 항목이 존재했으며, 무엇보다도 스캐너 간의 측정 기술을 비교하는 것이 불가능하였다. 왜냐하면, 각각의 자동 인체 측정 소프트웨어에서 사용하는 용어가 다를 뿐만 아니라 측정하고자 하는 영역과 방식도 서로 달라 비교 자체가 무의미했기 때문이다.

결론적으로 자동 랜드마크 측정을 위해서 오랜 기간 동안 많은 연구가 있었으나 다양한 체형에 대해 안정적으로 랜드마크를 찾는 방법은 아직도 개발되지 않았다. 그리고 현존하는 자동 인체 측정 소프트웨어들은 각 스캐너의 번들 프로그램으로 제공되기 때문에 다른 스캐너에서 얻은 데이터를 사용할 수 없으며, 각 소프트웨어에서 제공하는 랜드마크나 치수만을 측정할 수 있다. 그러나 사용자가 측정하고자 하는 치수를 사용하는 인체 측정 소프트웨어에서 제공하지 않을 수 있으며, 설령 측정하고자 하는 치수를 제공하더라도 사용자가 측정하고자 하는 치수의 정의와 소프트웨어가 제공하는 치수의 정의가 다를 수 있다. 결국, 사용자가 새로운 랜드마크나 치수를 측정하기 위해서는 매번 분석해야 할 인체 스캔 데이터의 특성에 맞게 사용자가 원하는 측정 방식으로 새롭게 프로그램을 개발하여야 하나, 대부분의 사용자는 프로그램을 개발하지 못할 뿐만 아니라 매번 새로운 프로그램을 만드는 것은 비경제적이다. 새로운 랜드마크나 치수를 측정하기 위한 다른 방법으로는 3차원 인체 형상 데이터에 수동으로 랜드마크를 찍고 경로를 만들어 측정할 수 있으나, 인체 형상 하나마다 일일이 수작업이 필요하여 시간이 많이 소요되고 비효율적이다. 따라서 스캐너의 종류와 관계없이 분석할 인체 측정 데이터의 특성에 맞게 사용자가 직접 랜드마크나 치수를 쉽게 정의하고 측정할 수 있는 범용 인체 측정 소프트웨어가 필요하다.

제 3 장 연구 방법

제 1 절 인체 모델 데이터 설계

시중에 상용화된 인체 스캐너의 종류는 다양하며, 각 스캐너에서 생성된 3차원 인체 스캔 데이터 역시 OBJ, PLY, STL 등의 다양한 형식이 존재한다. 그러나 다양한 인체 모델 데이터 형식을 분석할 수 있는 자동 인체 측정 소프트웨어가 아직 존재하지 않는다. 따라서 본 연구에서는 먼저 인체 측정 소프트웨어에서 사용할 간단한 형식의 3차원 인체 모델 데이터를 [그림 1]과 같이 설계하였다.



[그림 1] 3차원 모델의 데이터 구조

그리고 다양한 인체 모델 데이터 형식을 각각 분석하여 [그림 1]과 같은 3차원 인체 모델 데이터 구조로 변환하였으며, 이렇게 변환된 데이터를 UBM 형식이라 명명하였다. UBM 형식은 x, y, z값으로 이루어진 많은 점(**p**)과 이 점 중 세 점을 연결하여 삼각형(**f**)을 만드는데 필요한 점의 번호(**p**₀, **p**₁, **p**₂, ...)로 구성되며, 이는 삼각형 메쉬 구조를 이룬다.

제 2 절 인체 측정 소프트웨어 설계

본 연구에서는 사용자가 직접 랜드마크나 치수를 쉽게 정의하고 측정할 수 있도록 3차원 인체 측정 소프트웨어를 [그림 2]와 같이 설계하였다. 소스 코드의 재사용성, 유지보수성, 가독성이 용이하도록 인체 측정 소프트웨어를 Document-View 구조로 설계하여 데이터와 사용자 인터페이스를 분리하였다.

Document는 점 데이터 관리, 곡선 관리, 수식 해석, 3차원 모델 관리 등 다른 프로그램에서도 다양하게 활용될 수 있는 기본 클래스를 바탕으로 인체 모델과 측정을 총괄하는 TBodyMeasurement 클래스, 인체 단면 데이터를 관리하는 TBodySection 클래스, 랜드마크 데이터를 관리하는 TLandmarkList 클래스, 측정 결과 데이터를 관리하는 TbodyResult 클래스로 설계하였다.

View의 TMainForm 클래스는 프로그램의 메인 창으로 프로그램의 기본 바탕이며, 멀티 스크립트 편집, 측정 결과 표시, 인체 모델 수정 등의 추가 기능을 위해 ChildForm 클래스를 설계하였다. 또한, 시각적으로 확인하며 측정 작업이 가능하도록 인체 모델 데이터를 3차원으로 렌더링하기 위해 TRenderBase를 추가로 구성하였다.

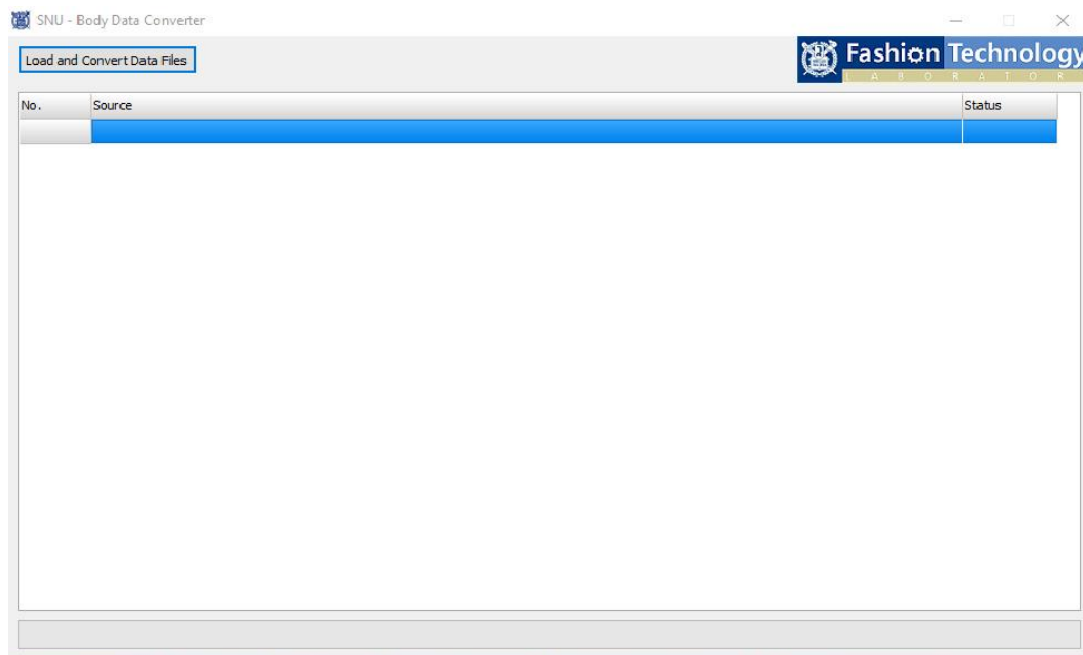


[그림 2] 인체 측정 소프트웨어 구조

제 4 장 연구 결과 및 고찰

제 1 절 데이터 변환 소프트웨어

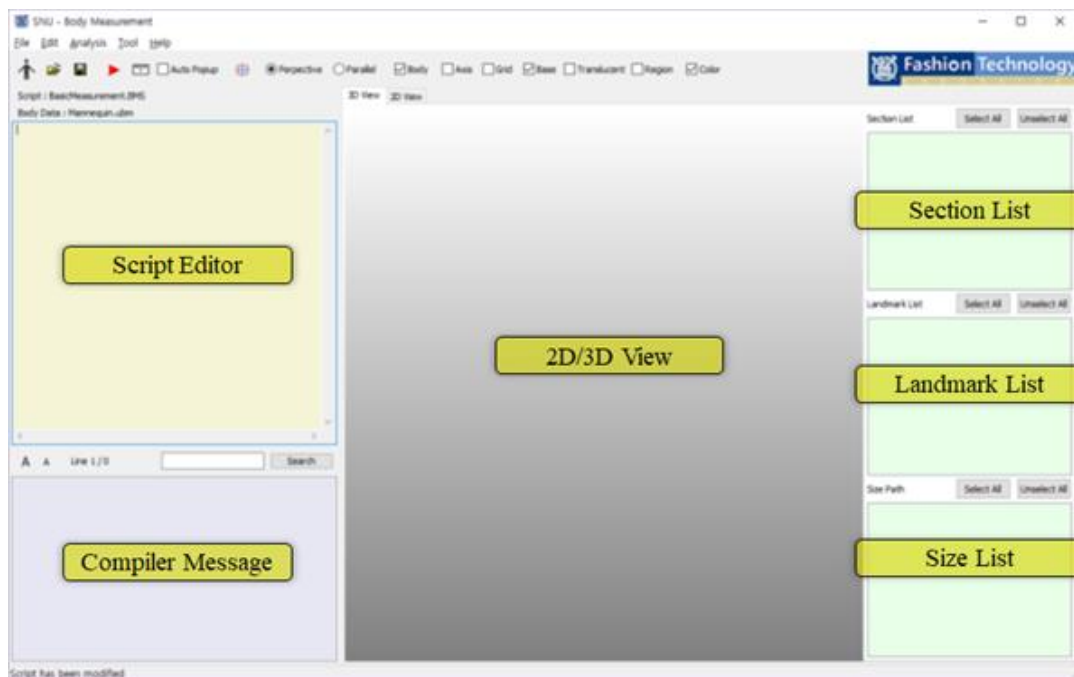
본 연구에서는 다양한 형식의 인체 모델 데이터 파일을 읽어 측정에 필요한 간단한 형식의 데이터로 변환하는 소프트웨어를 [그림 3]과 같이 개발하였다.



[그림 3] 데이터 변환 소프트웨어

제 2 절 통합 개발 환경

본 연구에서는 [그림 4]와 같이 인체 모델을 3차원으로 표시하고, 각종 단면의 형태를 2차원으로 표시하는 2D/3D 뷰(2D/3D View) 영역, 스크립트 언어를 작성하고 편집하는 스크립트 에디터(Script Editor) 영역, 실행 과정에서 각종 메시지를 표시하는 컴파일러 메시지(Compiler Message) 영역, 인체 모델의 각종 단면과 랜드마크, 그리고 치수 측정 결과를 리스트 형식으로 표시하는 단면/랜드마크/치수 리스트(Section/Landmark/Size List) 영역 등으로 이루어진 통합 개발 환경으로 인체 측정 시스템을 개발하였다.



[그림 4] 통합 개발 환경

분석하려는 인체 스캔 데이터를 지정하고, 측정하기 원하는 단면이나 랜드마크, 치수를 구하기 위해 스크립트 에디터 영역에 다양한 명령어를 조합하여 스크립트를 작성하고 이를 실행하면, 실행 경과를 컴파일러 메시지 영역에서 확인할 수 있다. 그리고 정상적으로 실행이 완료되면 2D/3D 뷰 영역에 지정한 모델과 함께 측정한 단면, 랜드마크, 치수가 시각적으로 표시되어 측정이 잘 이루어졌는지 확인할 수 있으며, 단면/랜드마크/치수 리스트에서 각 측정 항목을 확인할 수 있고, 측정 결과를 저장할 수도 있다. 비록 컴퓨터 프로그래밍 경험이 없는 사용자일지라도 이 모든 작업을 대화식 사용자 인터페이스를 이용하여 수행할 수 있는 통합 개발 환경으로 인체 측정 시스템을 개발하였다.

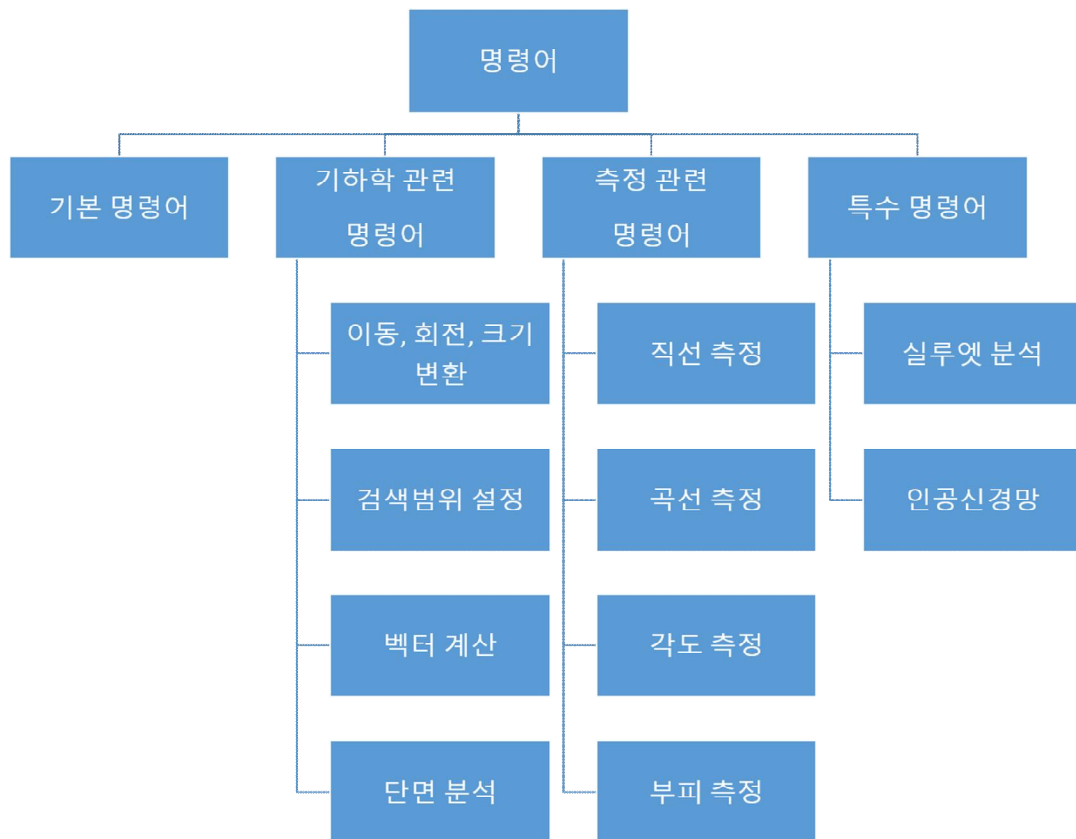
제 3 절 스크립트 시스템과 명령어 정의

본 연구에서는 간단한 프로그래밍 언어인 스크립트 언어로 개발하여 사용자가 직접 측정하고자 하는 랜드마크와 치수를 정의할 수 있게 했을 뿐만 아니라 미리 작성한 스크립트를 활용하면 많은 양의 인체 모델을 자동으로 측정할 수 있는 소프트웨어를 개발하였다.

스크립트 기반의 시스템은 소스 코드의 재사용성, 가독성, 유지보수성이 좋고 (Hargassner et al., 2008), 상호작용(interactive) 기반의 시스템보다 알고리즘 개발의 적합하며 공유하기 쉬운 장점이 있다 (Chaudhury et al., 2010).

스크립트 에디터에 측정하고자 하는 랜드마크와 치수를 측정하기 위한 스크립트를 작성하고 인체 모델을 지정한 후 스크립트를 실행하면 스크립트의 정해진 순서에 따라 인체 모델을 측정할 수 있다.

스크립트 명령어는 <명령어, 인자1, 인자2, ...>와 같은 형식으로 작성하며, 명령어는 [그림 5]와 같이 나눌 수 있다.



[그림 5] 명령어 분류

기본 명령어는 시스템 운영에 필요한 명령어이며, 기하학 관련 명령어로 인체 모델을 정렬하고, 검색 범위를 설정하거나 단면형상을 분석할 수 있다. 측정 관련 명령어는 기하학 관련 명령어로 찾은 랜드마크를 바탕으로 다양한 치수를 측정할 때 사용한다. 또한, 살점이나 겨드랑점과 같이 측정이 어려운 랜드마크를 찾기 위해 실루엣을 분석하여 랜드마크를 찾는 특수 명령어가 특별히 개발되었으며, 특히, 기하학 관련 명령어 조합만으로는 찾기 어려운 랜드마크를 측정하기 위해 인공신경망을 이용하여 랜드마크를 찾는 특수 명령어도 개발되었다. 주요 명령어에 대한 간단한 설명은 다음과 같으며, 전체 명령어는 [부록 1] 명령어 리스트를 통해 확인할 수 있다.

1. 기본 명령어

<Exit> 명령은 스크립트의 실행을 중단한다.

<Execute, FILE_NAME> 명령은 지정된 이름의 스크립트 파일을 실행하기 때문에 여러 인체 측정 프로젝트에서 공통적으로 수행되는 과정을 서브 루틴화하여 불러오는데 사용할 수 있다.

<Set, VARIABLE_NAME, EQUATION> 명령은 지정된 이름의 변수에 수식의 변환한 값을 대입한다. 수식은 숫자, 변수를 이용한 사칙연산을 써서 표현할 수 있다.

2. 기하학 관련 명령어

2.1. 이동, 회전, 크기 변환

인체 스캔 데이터가 항상 올바르게 정렬되어 있는 것이 아니라 인체 스캐너 기기에 따라 중심에 인체 모델이 없거나, 회전되어 있거나, 스케일이 다를 수 있다. 따라서 인체 모델을 이동하거나, 회전하거나, 또는 스케일을 조정할 수 있는 명령어가 필요하다.

<Centering> 명령은 모델의 중심을 좌표축의 중심으로 이동하는 명령이다. 경우에 따라 X, Y, Z축 중 선택한 축만을 기준으로 이동할 수 있다. <Centering, XYZ> 라고 쓰면 모든 축을, <Centering, X> 또는 <Centering, YZ> 라고 쓰면 X축이나 YZ축만을 기준으로 삼아 중심으로 이동할 수 있다.

<Move> 명령은 X, Y, Z축을 따라 주어진 좌표만큼 모델을 이동한다.

<Rotate, ANGLE_X, ANGLE_Y, ANGLE_Z> 명령은 X, Y, Z축을 중심으로 주어진 각도만큼 모델을 회전한다.

<Scale, RATIO> 명령은 인체 모델을 주어진 배율만큼 확대하거나 축소한다.

2.2. 측정

<GetSize> 명령은 모델의 접하는 직육면체를 만들어 모델의 크기를 계산한 뒤 각각의 내부 변수 Width(X축), Height(Y축), Depth(Z축),

Top(max. y), Bottom(min. y)에 해당 값을 저장한다.

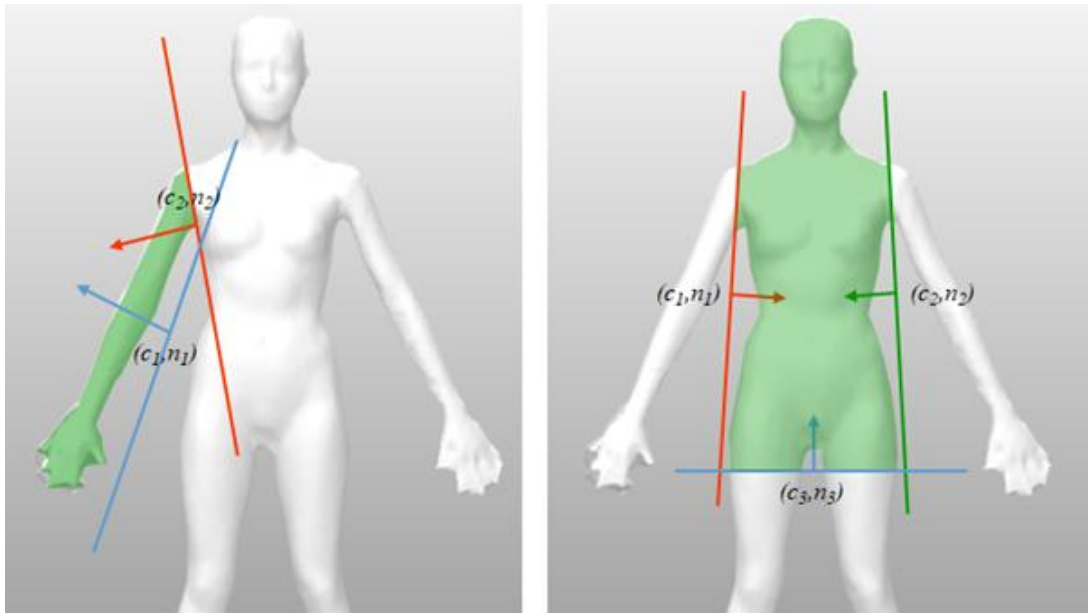
<GetDistance, VARIABLE_NAME, POINT1, POINT2> 명령은 두 점 사이의 거리를 계산하여 지정한 변수 이름으로 저장한다.

<.> 연산자는 점의 좌표를 조회할 때 사용할 수 있다. 예를 들어, P라는 점이 있을 때, P.x와 같은 표현으로 P점의 x좌표를 조회할 수 있다.

2.3. 검색 범위 설정

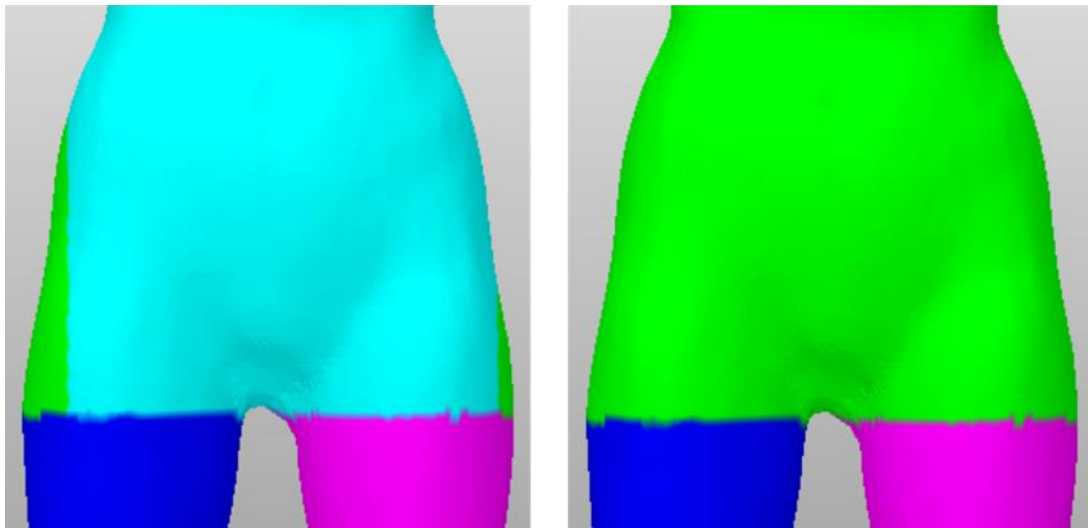
<SetHeightRange, RANGE_NAME, LOW, HIGH> 명령은 이후 수행할 각종 검색 과정에서 랜드마크를 검색하는 Y축 범위를 설정한다. 인체 모델에서 랜드마크를 측정할 때 인체 모델 샘플의 통계 데이터로부터 얻은 일정 범위에서 검색하는 방법이 널리 쓰인다. LOW, HIGH값을 키(Height)에 대한 비율로 설정하면 지정한 범위명으로 저장된다. 예를 들어, <SetHeightRange, CrotchRange, 0.4, 0.5> 이라고 설정한 후 CrotchRange 안에서 살점을 찾는 명령을 실행하면, y값의 범위가 (키 × 0.4) ~ (키 × 0.5)에 해당하는 점들 안에서만 살점을 검색한다. 또한 <.> 연산자를 써서 CrotchRange.min, CrotchRange.max와 같이 해당 범위의 최솟값과 최댓값을 조회할 수도 있다.

<SetRegion> 명령은 <SetRegion, REGION_NAME, CENTER1, NORMAL1, CENTER2, NORMAL2, ...>와 같은 형식을 가지며, [그림 6]에 보인 것과 같이 중심점과 법선 벡터로 정의되는 평면 위 공간들의 교집합으로 영역을 설정한다. 이와 같이 설정된 영역은 SetHeightRange에서 정한 영역과 마찬가지로 랜드마크를 찾는 명령의 검색 범위가 된다.



[그림 6] 영역 정의를 위한 도식화

그리고 $\langle \text{UnionRegion, REGION_NAME, REGION1, REGION2, } \dots \rangle$ 명령을 이용하여 [그림 7]와 같이 여러 영역의 합집합으로 새로운 영역을 정의할 수도 있으며, 새롭게 정의된 영역 역시 검색 영역으로 설정할 수 있다.



[그림 7] UnionRegion의 예

또한, 검색 영역을 지정하는 명령으로 <SetSearchRange>이 있다. <SetSearchRange, REGION_NAME, min_x, min_y, min_z, max_x, max_y, max_z> 명령을 사용하면 육면체 형태의 검색 영역도 지정할 수 있다.

2.4. 벡터 계산

<SetPoint, POINT_NAME, x, y, z> 명령은 x, y, z 좌표를 갖는 주어진 이름의 점을 생성한다. 점의 좌표를 수식으로 표현하는 것도 가능하다.

<SetPointDivide, POINT_NAME, POINT1, POINT2, m, n> 명령은 두 점 사이를 m : n으로 내분하는 점의 좌표로 새 점을 생성한다.

<GetDirection, POINT_NAME, POINT1, POINT2> 함수는 POINT1에서 POINT2로 향하는 방향 벡터의 좌표로 새 점을 생성한다.

<RotatePoint, POINT, ANGLE_X, ANGLE_Y, ANGLE_Z> 명령은

지정된 점을 X, Y, Z축 둘레로 지정한 각도만큼 회전한다.

2.5. 단면 분석

대부분의 인체 측정은 단면형상으로부터 이루어지기 때문에 단면을 정의하고 분석하는 것은 매우 중요하다. 따라서 다양한 단면을 지정하고 분석할 수 있는 여러 명령어를 다음과 같이 개발하였다.

<GetSection, SECTION_NAME, POINT, NORMAL> 명령은 POINT를 지나고 법선 벡터가 NORMAL인 평면으로 인체 모델을 절단한 단면을 생성한다.

<GetSection3P, SECTION_NAME, POINT1, POINT2, POINT3> 명령은 3개의 점으로 이루어지는 평면으로 인체 모델을 절단한 단면을 생성한다.

추출된 단면형상은 자동적으로 XY 평면에 투영되며, 이때 단면의 법선 벡터가 Y축과 평행이 되도록 좌표를 변환한다. 이렇게 투영된 단면에 대해 SECTION.width / .height / .girth 와 같은 연산자를 사용하여 단면의 폭(X축), 높이(Y축), 그리고 둘레를 구할 수 있으며, 2D 뷰 영역을 통해 추출된 단면의 형상을 확인할 수 있다.

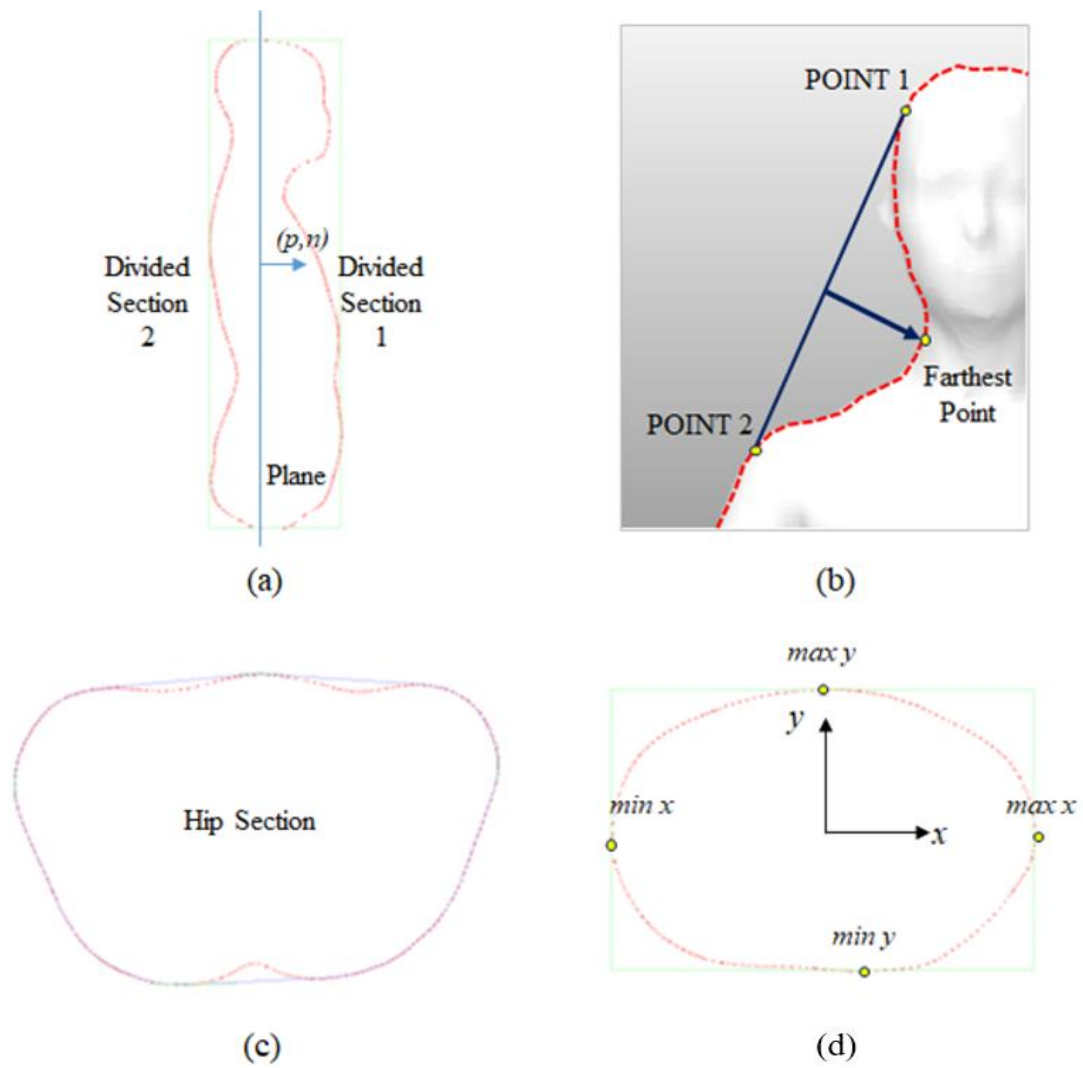
<GetSectionByGirth, SECTION_NAME, REGION_NAME, MAX/MIN, HEIGHT_RANGE> 명령은 지정한 영역 내의 HeightRange 범위 안에서 Y축 방향으로 1mm 간격마다 수평 단면을 획득하여 단면의 둘레를 구하고, 그 둘레의 길이가 가장 길거나(MAX)/짧은(MIN) 단면을 구하는 명령어로, 예를 들어, 토르소에서 둘레의 길이가 가장 짧은 곳을 허리로 지정하고자 할 때, 이 명령어를 이용하면 허리 단면을 쉽게 획득할 수 있다.

<SetPlane, PLANE_NAME, POINT, NORMAL> 명령은 POINT를 지나며 법선 벡터가 NORMAL인 평면을 정의하는 명령이다. 이 명령은 <GetSection> 명령과 달리 실제로 인체와 평면과의 단면을 구하지 않으며, 다만 평면의 방정식을 하나 생성할 뿐이다. 예를 들어, 이렇게 정의된 평면은 <DivideSection, SECTION_NAME, PLANE, DIVIDED1, DIVIDED2> 명령에 사용할 수 있다. 이 명령은 하나의 지정한 단면을 평면을 기준으로 둘로 나누어준다. 예를 들어 [그림 8(a)]와 같이 인체의 종단면을 얻었을 때, 한 평면을 기준으로 이 단면을 DIVIDE1, DIVIDE2 단면으로 나눠서 각각 다른 연산에 사용하는 것이 가능하다.

<GetFartestPoint, POINT_NAME, SECTION, POINT1, POINT2> 명령은 주어진 단면에서 점 POINT1과 POINT2를 연결하는 선분과 가장 멀리 떨어진 점을 구하는 명령이다. 이 명령은 [그림 8(b)]와 같이 목 옆점을 찾을 때 사용할 수 있다.

<GetConvexHull, SECTION1, SECTION2, ...> 명령은 [그림 8(c)]와 같이 주어진 단면들에 대해 단면을 둘러싸는 convex hull을 생성한다.

<GetMaxPeak2D, POINT, SECTION, X or Y> 명령은 [그림 8(d)]에 보인 것과 같이 평면에 투영된 단면에서 X축 또는 Y축 방향의 좌표가 최대인 점을 찾는다. 유사한 방법으로 <GetMinPeak2D, POINT, SECTION, X or Y> 명령은 평면에 투영된 단면에서 X축 또는 Y축 방향의 좌표가 최소인 점을 찾는다.



[그림 8] 단면 분석의 예

- (a) 단면 분리 (b) Farthest Point 찾기
(c) Convex hull 생성 (d) Peak Point 찾기

3. 측정 관련 명령어

단면 분석을 통해 측정한 값들을 시각적으로 표시하거나 랜드마크 사이의 직선이나 곡선 패스를 생성하고 이를 측정하고 각도, 특정 영역의 부피 등을 측정하기 위해 다음과 같은 명령어를 개발하였다.

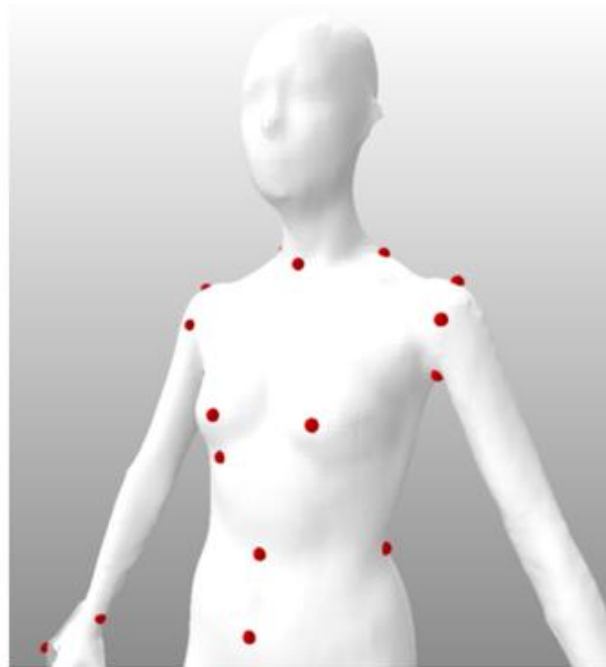
3.1. 랜드마크 및 측정 결과

<SetResult, RESULT_NAME, VALUE> 명령은 주어진 이름으로 결과 리스트에 값을 추가하며, 이 리스트는 측정이 종료되면 [그림 9(a)]와 같이 표로 표시되므로 측정한 치수를 확인할 때 사용하며, 이 측정 결과를 저장할 수도 있다.

<SetLandmark, NAME, POINT or [x y z]> 명령은 POINT를 이용하거나 x, y, z 좌표를 써서 새로운 랜드마크를 정의한다. 랜드마크는 [그림 9(b)]와 같이 인체 모델 상에 빨간색 구 형상으로 표시된다.

Measurement Result		
Name	mm	in
Height	1621	63.8
Abdomen Height	913	36.0
Back Waist Height	1008	39.7
BP Height	1163	45.8
Ab to BP	250	9.9
BP Width	125	5.0
Chest Depth	17	0.7
Neck to BP	211	8.3
TorsoDepth	214	8.4
TorsoHeight	891	35.1
TorsoGirth	1995	78.5
Path1	294	11.6
Path2	576	22.7
Path3	946	37.3
Path4	353	13.9

(a)



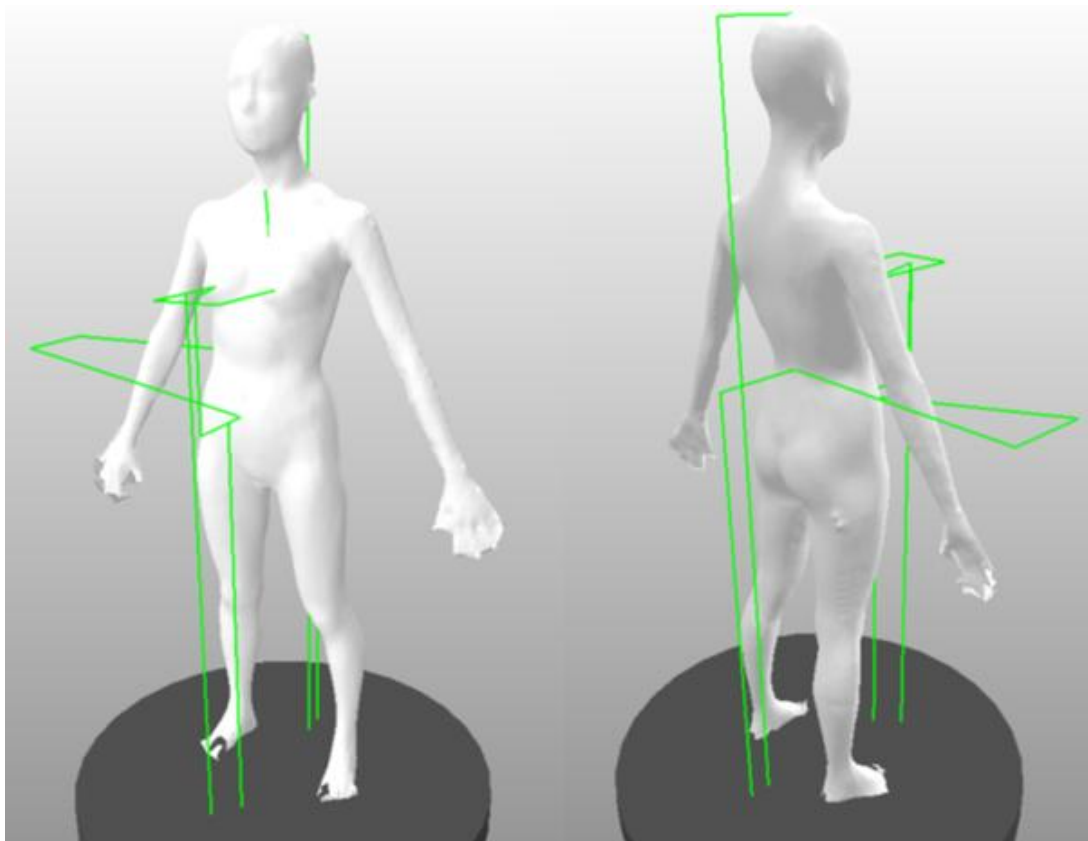
(b)

[그림 9] 측정 결과 예시
(a) 결과 리스트 (b) 랜드마크

3.2. 직선 측정

<SetHeightResult, RESULT_NAME, POINT1, (POINT 2)> 명령은 POINT1, POINT2가 주어졌을 때는 두 점 간의 높이 차이(y좌표)로 새로운 측정치를 정의하고 측정 결과를 결과명에 저장한다. POINT1만 주어진 경우에는 바닥면에서 점까지의 높이로 측정치를 정의하고 측정 결과를 저장하며, 비슷한 방법으로 <SetWidthResult, RESULT_NAME, POINT1, POINT2> 명령은 두 점 사이의 폭(x좌표)의 차이, <SetDepthResult, RESULT_NAME, POINT1, POINT2> 명령은 두 점 사이의 깊이(z좌표)의 차이로 측정치를 정의하고 측정 결과를 저장한다.

<SetDistanceResult, RESULT_NAME, POINT1, POINT2> 명령은 두 점 사이의 직선거리로 측정치를 정의하고 측정 결과를 저장한다. 측정이 잘 이루어졌는지 확인하기 위해 측정치를 시각화한 예는 [그림 10]에 보인 것과 같다.



[그림 10] 직선 측정 시각화 예시

3.3. 곡선 측정

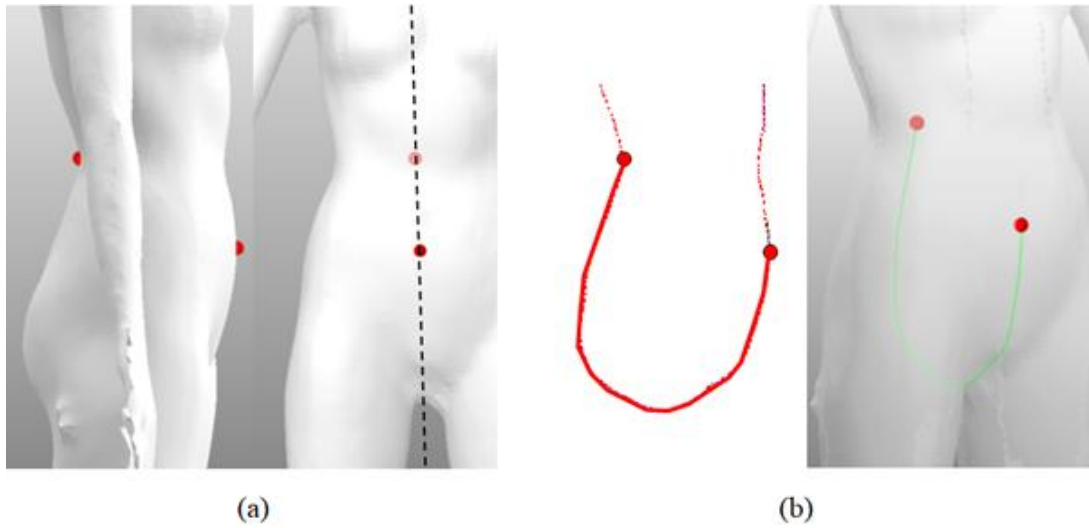
실제 인체를 측정할 때는 두 점 간의 거리를 재는 것보다 인체 표면을 따라가면서 컨벡스(convex)한 경로를 재는 경우가 더 많기 때문에 이런 측정을 위한 기능이 필요하다. 따라서 이런 경로를 패스(path)라고 정

의하였으며, 본 연구에서는 패스를 구하는 다양한 방법을 개발하였다.

<SetSectionPath, PATH_NAME, SECTION_NAME> 명령은 주어진 단면의 컨벡스 헐(convex hull)을 측정 경로로 추가한다.

<SetPathResult3, NAME, POINT1, POINT2, POINT 3> 명령은 공간상의 세 점을 지나는 평면으로 인체를 잘라 단면을 구하고, 단면에서 세 점을 통과하는 경로만의 컨벡스 헐을 구한 후 POINT1에서 POINT3까지의 경로를 패스로 추가한다.

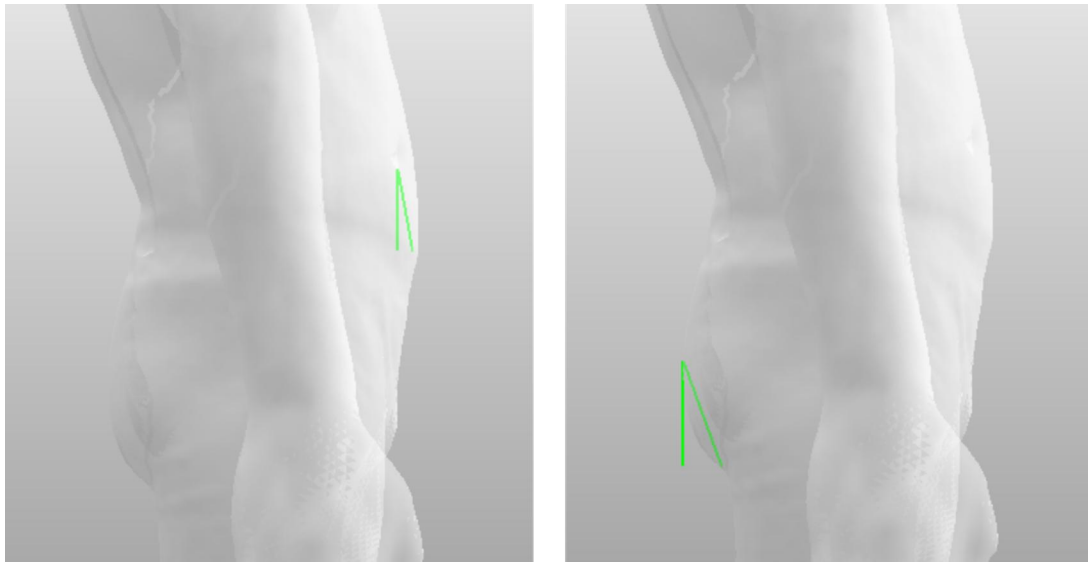
그런데 대부분의 인체 치수는 시작점과 끝점 두 점만을 지정하고 측정하는 경우가 많기 때문에 <SetPathResult2, PATH_NAME, POINT1, POINT2, Front/Side/Top> 명령을 개발하였다. 이 명령은 POINT1, POINT2를 지나며 동시에 앞(Front), 옆(Side) 또는 위(Top)에서 인체 모델을 봤을 때 직선으로 보이는 평면으로 인체를 자른 단면을 구한 다음, 두 점 사이의 점들로 컨벡스 헐을 구성하고 이를 경로로 설정한다. [그림 11]는 배돌출점과 허리뒤점 사이의 패스를 구하는 예이다. 먼저 [그림 11(a)]와 같이 두 점을 지나면서 정면에서 봤을 때 직선이 되는 평면을 정의한 뒤 이것으로 인체의 단면을 구하고 두 점 사이의 점들로 컨벡스 헐을 만든 결과는 [그림 11(b)]와 같다.



[그림 11] 두 점 사이의 패스 정의 사례
(a) 자르는 단면 정의 (b) 패스 시각화

3.4. 각도 측정

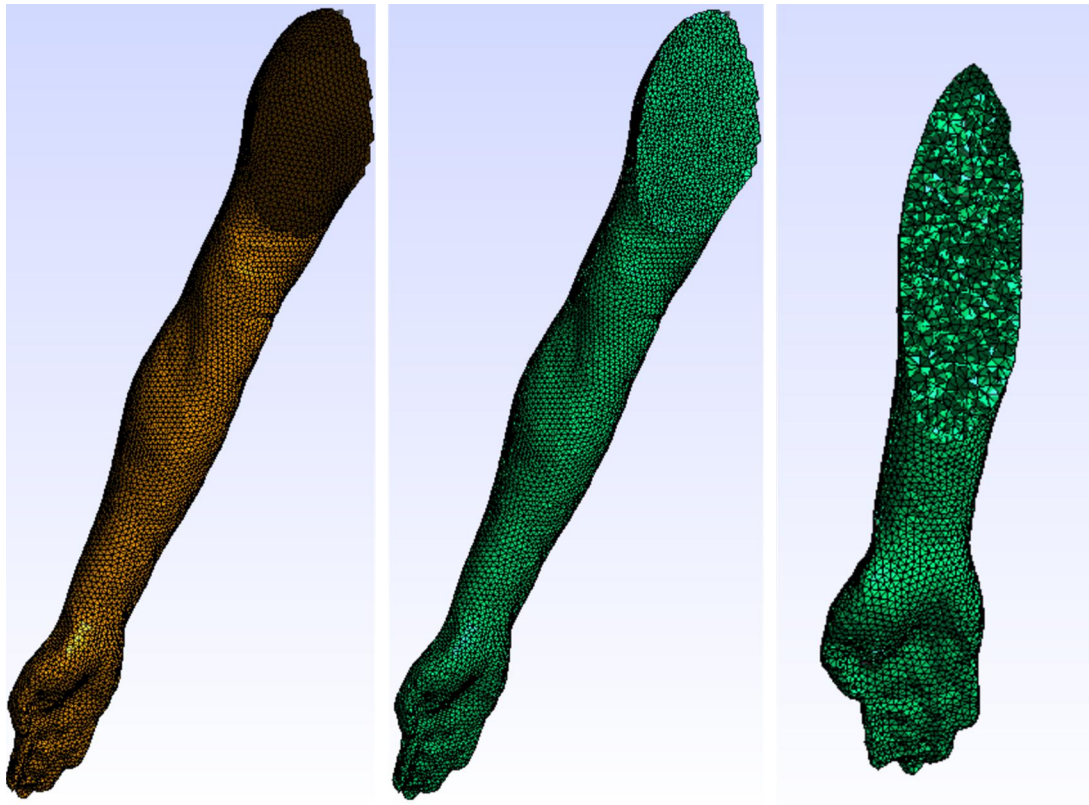
3차원 인체 스캐너로부터 얻은 인체 모델 데이터를 분석하기 위해서는 길이뿐만 아니라 배상부각, 엉덩이하부각과 같은 각도 역시 측정 가능하여야 한다. 본 연구에서는 두 개의 랜드마크(또는 점)와 방향 벡터로 각도를 측정할 수 있는 명령어를 개발하였다. <SetAngleResult, RESULT_NAME, POINT1, POINT2, VECTOR, Front/Side/Top> 명령은 점1에서 점2로 향하는 벡터와 지정한 방향 벡터 사이의 각을 앞, 옆, 또는 위에서 인체 모델을 봤을 때의 각의 크기를 계산하여 결과에 추가하고, [그림 12]과 같이 3D View 영역에 시각적으로 나타낸다.



[그림 12] 각도 측정 사례

3.5. 영역 부피 측정

길이, 각도뿐만 아니라 영역의 부피를 측정할 수 있는 명령어도 개발하였다. <SaveRegionAsMSH, REGION_NAME> 명령은 부피를 측정하기 위한 사전 준비 작업으로, 부피를 측정하기 원하는 영역을 지정하고 명령어를 실행하면 인체모델_영역명의 메쉬 파일이 [그림 13(a)]와 같이 생성된다. <CalculateVolume, MESH_FILE_NAME> 명령을 실행하면 지정된 메쉬 파일을 [그림 13(b)(c)]와 같이 메시 내부의 빈 공간을 사면체로 완전히 메운 3차원 볼륨 메쉬를 생성하고, 생성된 개별 사면체 부피의 총합으로 영역의 부피를 계산하며 (Geuzaine and Remacle, 2009), 계산 결과는 컴파일러 메시지 영역에 출력된다.



(a)

(b)

(c)

[그림 13] 부피 계산을 위한 볼륨 메쉬 생성 사례

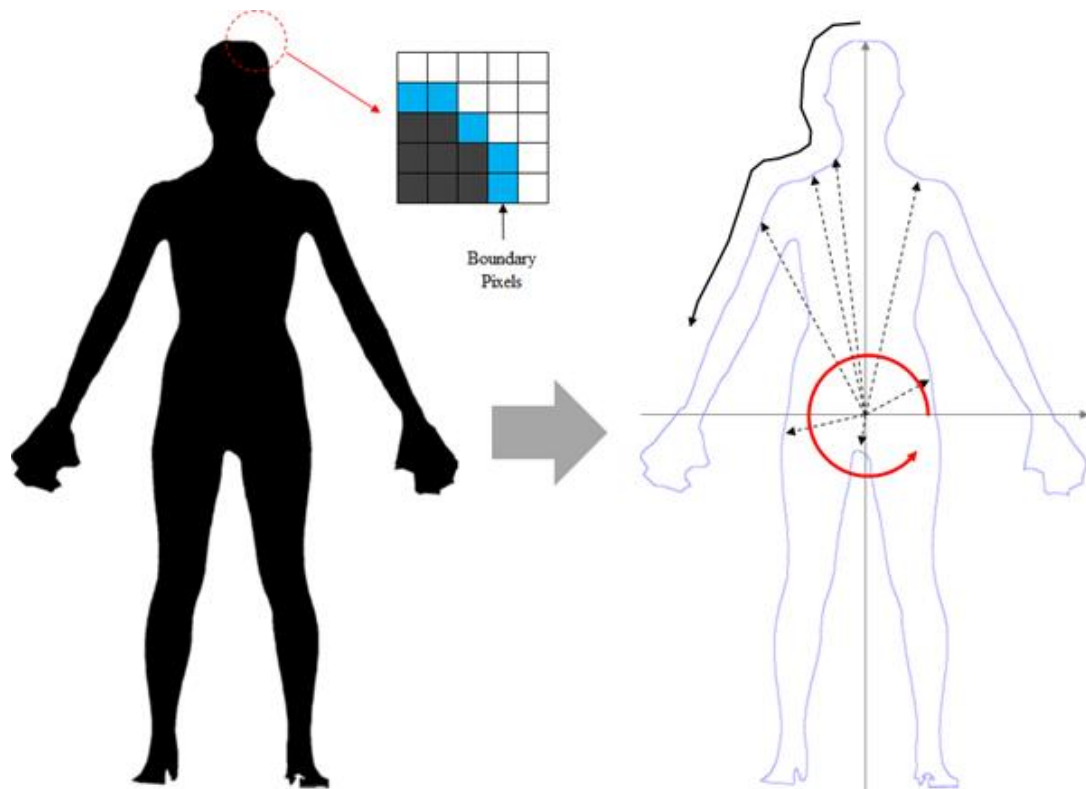
(a) 볼륨 메쉬 생성 전 (b) 볼륨 메쉬 생성 후 (c) 볼륨 메쉬 내부 단면

4. 특수 명령어

4.1. 실루엣 추출을 이용한 살점, 겨드랑점 찾기

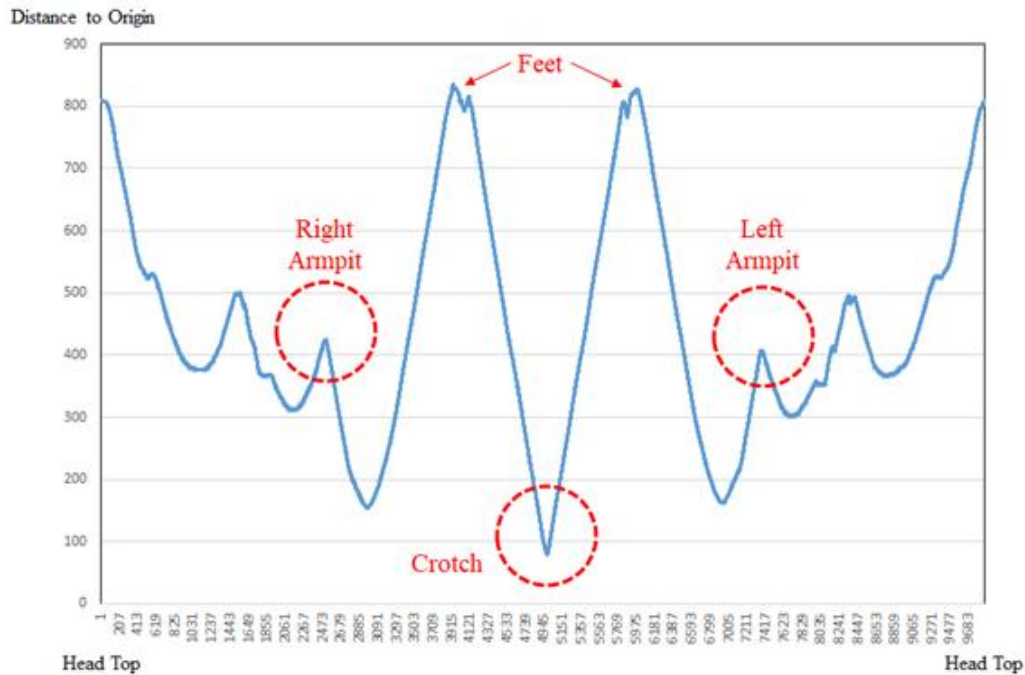
앞에서 정의한 여러 기하학 관련 명령어로도 측정이 곤란한 랜드마크들이 다수 있는데, 측정의 기본이 되는 살점과 겨드랑점도 그에 속하는 랜드마크이다. 이 랜드마크들은 몸통, 팔, 다리를 구분하는데 필요한 매우 중요한 랜드마크이기 때문에 본 연구에서는 이 랜드마크를 찾을 수 있는 특별한 명령어를 개발하였다.

먼저 [그림 14]와 같이 인체의 정면 투영 이미지를 만든 후 이 이미지의 외곽선만을 추출하고, 이 점들을 반시계방향으로 연결하여 회전하는 실루엣 곡선을 추출한다.



[그림 14] 인체 실루엣 분석 도식화

이 곡선을 머리마루점부터 시작하여 반시계방향으로 실루엣 상의 모든 점을 한 바퀴 돌면서 원점까지의 거리를 측정해서 그래프로 나타낸 것은 [그림 15]와 같다.



[그림 15] 각 실루엣 점으로부터 원점까지의 거리 측정 그래프

[그림 15]에서 볼 수 있는 것과 같이 실루엣 점에서 원점까지의 거리는 살점에서 가장 짧고, 머리끝과 발끝에서 가장 길다. 또한, 살점을 중심으로 좌우로 진행했을 때, 살점으로부터 두 번째 높은 피크가 겨드랑점을 나타낸다는 것을 알 수 있다. 그러나 발의 형상이 복잡하여 발 영역에서 여러 개의 피크가 나타나기 때문에 단순히 두 번째 피크를 찾는 것으로 겨드랑점을 찾을 수 없다. 이 문제를 해결하기 위해 여러 인체 데이터를 이용하여 테스트를 해본 결과 두 번째 피크가 나타나는 대략적인 범위를 알 수 있었으며, 이를 이용해서 겨드랑점의 위치를 찾을 수 있다. 일단, 살점과 겨드랑점의 실루엣 상에서의 위치를 찾은 후 3차원 인체 모델을 구성하는 점 중에서 이 점과 가장 가까운 점을 찾아서 랜드마크로 지정하였다. 이 복잡한 과정을 <FindCrotch, CROTCH_LANDMARK_NAME>과 <FindArmpit, LEFT_ARMPIT_LANDMARK

_NAME, RIGHT_ARMPIT_LANDMARK_NAME> 명령으로 간단히 수행할 수 있도록 특수 명령어를 개발하였다.

4.2. 인공신경망을 이용한 랜드마크 찾기

앞에서도 언급했듯이 비접촉으로 측정하는 인체 스캐너의 특성상 목뒤점과 같이 직접 만져보거나 뼈를 통해 검색할 수 있는 랜드마크는 기하학 관련 명령어의 결합만으로는 측정하기가 어렵다. 따라서 본 연구에서는 스크립트 작성으로 미리 찾은 랜드마크를 이용하여 기계 학습에서 얻은 인공신경망으로 랜드마크를 찾는 특수 명령어를 개발하였다.

<LoadANN, NETWORK_NAME, FILE_NAME> 명령어를 통해 미리 인공신경망으로 기계 학습한 파일을 프로그램 상에 네트워크명으로 로드한다. <GetANNPoint, NETWORK_NAME, LANDMARK_NAME, POINT1, POINT2, ...> 명령어로 LoadANN 명령어로 불러온 네트워크와 학습에 사용된 점을 모두 지정하면 인공신경망을 통해 랜드마크를 검색할 수 있다.

인공신경망을 이용한 랜드마크 검색을 검증하기 위해 명령어 결합으로 찾은 특정 랜드마크를 이용하여 검색하기 어려운 랜드마크를 찾는 실험을 진행하였다. 먼저, 120명의 인체 모델 데이터에서 목앞점, 젖꼭지점, 겨드랑점과 같은 랜드마크를 검색하였다. 다음으로 기하학 관련 명령어의 결합만으로는 측정하기 어려운 랜드마크 중 시각적으로 구별이 용이한 목뒤점과 배꼽점을 수동으로 표시하였다. 이와 같은 방식으로 랜드마크를 얻은 120개의 인체 모델 데이터 중 110개는 학습에 사용하였으며, 남은 10개는 검증에 활용하였다. 끝으로, 기계 학습으로 얻은 인공신경망 데이터를 활용하여 120개 인체 모델 데이터 이외에

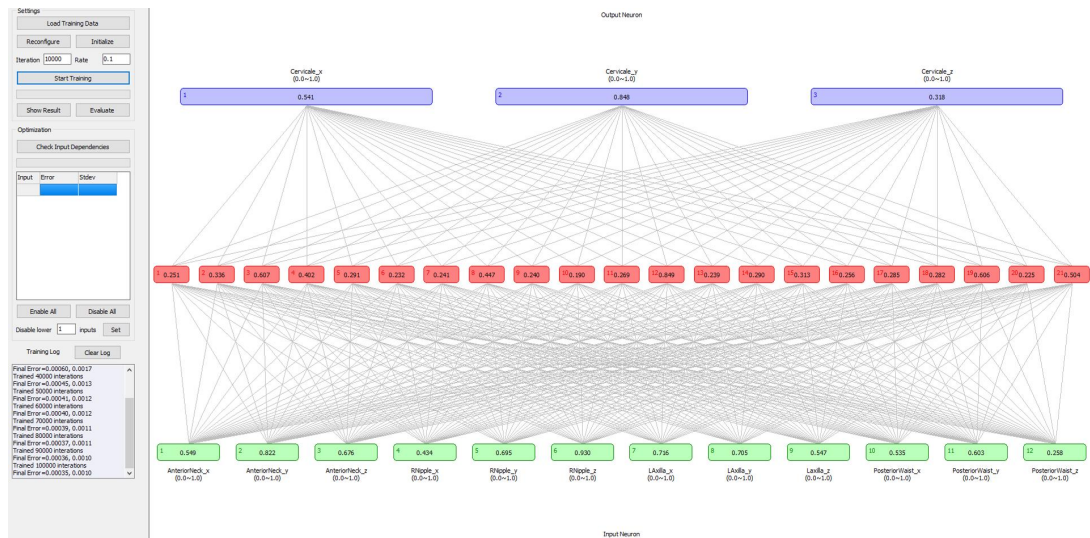
다른 인체 모델 데이터에서 목뒤점과 배꼽점을 검색할 수 있는지를 실험하고, 이를 시각적으로 검증했다.

[표 1]은 목표 랜드마크를 찾기 위해 사용된 입력한 랜드마크들을 보여주는 기계 학습 데이터이다.

[표 1] 기계 학습 데이터

입력 랜드마크	목표 랜드마크	입력 랜드마크	목표 랜드마크
목앞점	목뒤점	배돌출점	배꼽점
오른 젖꼭지점		허리뒤점	
왼 겨드랑점		오른 겨드랑점	
허리뒤점		목앞점	
-		왼 젖꼭지점	

입력 랜드마크를 다양하게 조합하여 실험을 진행하였으며, 그 중 학습 결과가 가장 좋은 입력 랜드마크의 조합을 선택하였다. 이 입력 랜드마크들은 다른 랜드마크에 비해 목표 랜드마크를 잘 나타내는 랜드마크라고 할 수 있다.



[그림 16] 인공신경망 학습 사례

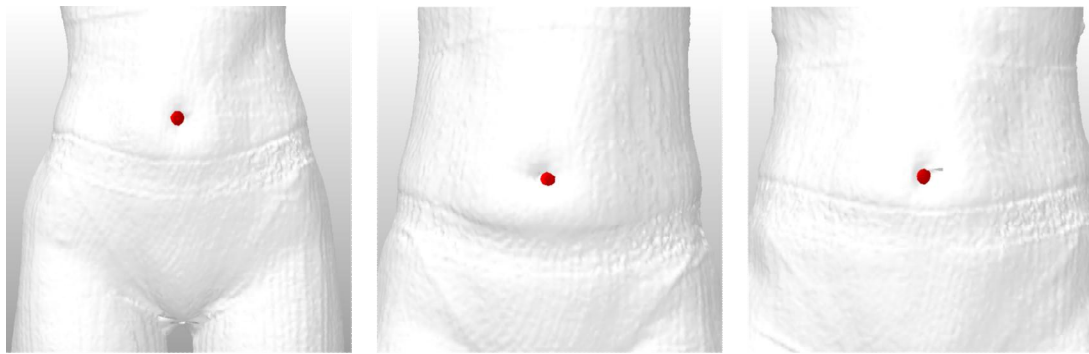
[그림 16]은 목뒤점을 찾기 위해 구성한 인공신경망이다. 랜드마크 4개의 좌표 12개가 입력되었으며, 히든 레이어는 1층, 21개의 뉴런으로 구성하였다. 결과 랜드마크는 목뒤점이며 3개의 좌표로 이루어진다. 학습 속도 0.1로 10,000회씩 나누어 총 100,000회 학습시켰으며, 최종 에러는 평균 0.00035, 표준오차 0.0010이 발생했다.

[표 2]는 목뒤점과 배꼽점을 찾기 위해 위와 같은 방식으로 인공신경망을 구성하고 기계 학습시킨 결과를 정리한 표이다.

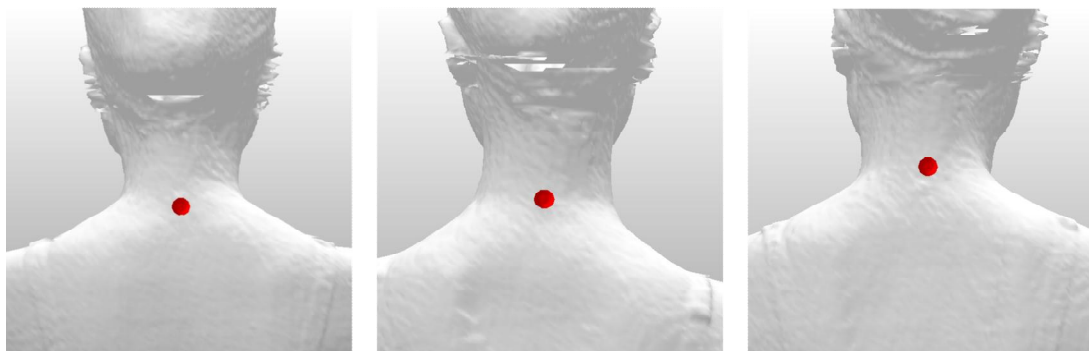
[표 2] 인공신경망 구성

구 분	목뒤점	배꼽점
입력 뉴런	12개	15개
히든 레이어	1층	1층
히든 뉴런	21개	24개
출력 뉴런	3개	3개
학습 횟수	100,000번	100,000번
학습 속도	0.1	0.1
최종 에러(평균, 표준편차)	0.00035, 0.0010	0.00055, 0.0016

위와 같이 인공신경망을 구성하고 기계 학습을 통하여 얻은 데이터를 학습에 사용된 120개 인체 모델 데이터를 제외한 다른 인체 모델 데이터에 적용하여 목뒤점과 배꼽점을 찾은 결과는 [그림 17]과 같다.



(a)



(b)

[그림 17] 인공신경망을 활용한 랜드마크 측정 결과
(a) 배꼽점 (b) 목뒤점

[그림 17]에서 볼 수 있듯이 다양한 모델에서 안정적으로 목뒤점과 배꼽점이 찾아지는 것을 알 수 있다. 이와 같은 방식으로 인공신경망을 구성하고 기계 학습을 활용한다면 목뒤점과 배꼽점뿐만 아니라 찾기 어려운 다양한 랜드마크 역시 검색 가능할 것으로 기대된다.

제 4 절 스크립트 작성

원하는 랜드마크를 찾거나 치수를 측정하기 위해 앞에서 설명한 다양한 명령어를 결합하여 스크립트를 작성할 수 있다. 측정을 위한 주요 스크립트 작성 예는 다음과 같다.

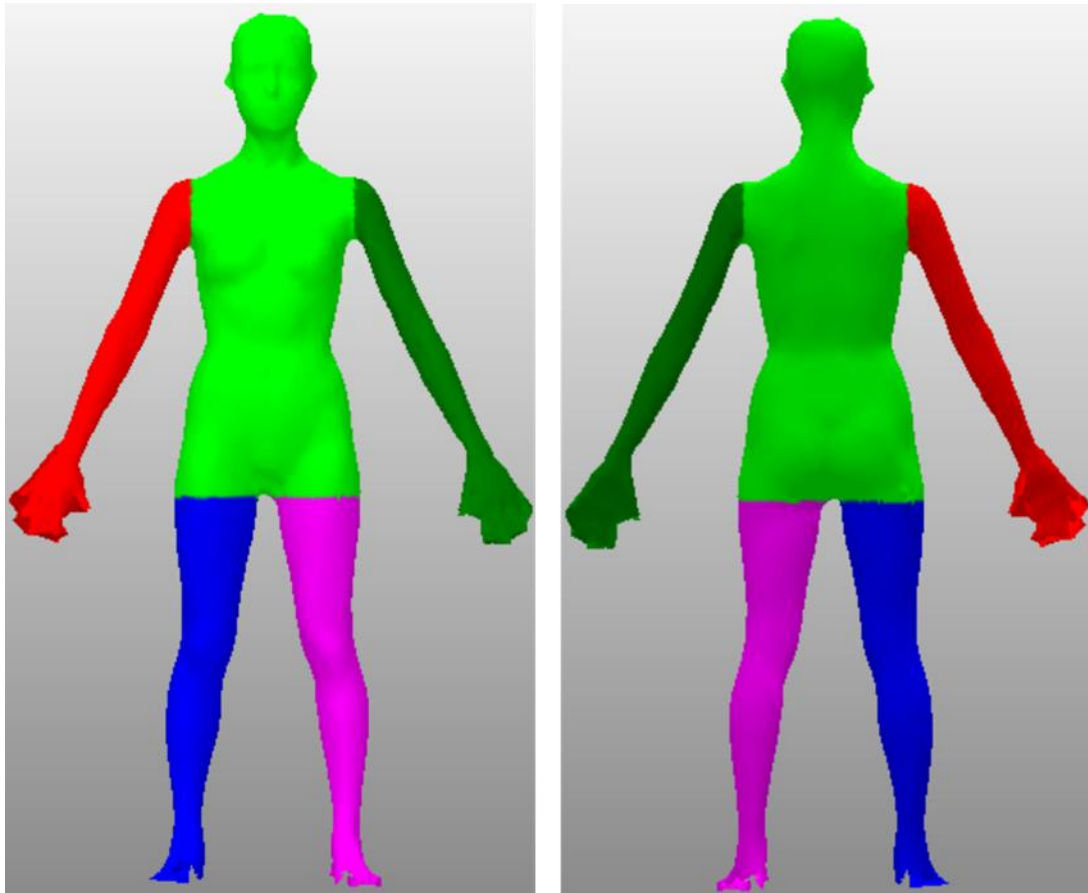
1. 인체 영역 분할(Body Separation)

인체 영역을 몸통과 팔, 다리로 분할하고 검색 영역을 한정하는 과정은 다양한 랜드마크 및 치수를 측정하기 위해 매우 유용하다. 먼저, 영역 분할을 위해서는 분할의 기준이 되는 점과 방향 벡터가 필요하다. 이때, 기준이 되는 점은 살점과 겨드랑점, 그리고 팔과 몸통 사이의 임의의 점이다. 살점과 겨드랑점은 특수명령어인 FindCrotch와 FindArmpit를 통해 찾을 수 있으며, 팔과 몸통 사이의 임의의 점은 새끼손가락 끝점과 살점의 중점(O)을 찾아 이용하였다. 방향 벡터는 기본적으로 왼쪽 $[-1 \ 0 \ 0]$, 오른쪽 $[1 \ 0 \ 0]$, 위 $[0 \ 1 \ 0]$, 아래 $[0 \ -1 \ 0]$ 를 이용했으며, 겨드랑점에서 중점(O)으로 향하는 벡터의 수직 벡터와 수직 벡터의 반대 방향 벡터를 구했다. 끝으로 SetRegion 명령어를 사용하여 각 영역의 교집합으로 몸통, 팔, 다리 영역을 분리했으며, 이를 수행한 자세한 스크립트는 [표 3]과 같다.

[표 3] 인체 영역 분할을 위한 스크립트

스크립트	실행 내용
FindCrotch,Crotch FindArmpit,LAxilla,RAxilla GetMinXPoint,RHand GetMaxXPoint,LHand	살점 찾기 겨드랑점 찾기 오른손 끝점 지정 왼손 끝점 지정
SetPointDivide,RO,RHand,Crotch,1,1 SetPointDivide,LO,LHand,Crotch,1,1	중점(O) 설정
SetPoint,LeftD,[-1 0 0] SetPoint,RightD,[1 0 0] SetPoint,DownD,[0 -1 0] SetPoint,UpD,[0 1 0]	Left, Right, Down, Up 방향벡터 지정
GetDirection,RR,RAxilla,RO GetNormal,RRNZ,RR,Z SetPointInverse,RRNZi,RRNZ GetDirection,LL,LAxilla,LO GetNormal,LLNZi,LL,Z SetPointInverse,LLNZ,LLNZi	겨드랑점과 중점을 이용하여 방향벡터 지정
SetRegion,RightArm,RAxilla,LeftD,RO,RRNZ SetRegion,LeftArm,LAxilla,RightD,LO,LLNZ SetRegion,RightLeg,Crotch,DownD,Crotch, LeftD,RO,RightD SetRegion,LeftLeg,Crotch,DownD,Crotch, RightD,LO,LeftD SetRegion,Chest1,Crotch,UpD,RO,RRNZi,LO, LLNZi SetRegion,Chest2,Crotch,UpD,RAxilla,RightD ,LAxilla,LeftD UnionRegion,Chest,Chest1,Chest2 DeleteRegion,Chest1,Chest2	오른팔 영역 지정 왼팔 영역 지정 오른다리 영역 지정 왼다리 영역 지정 몸통1 영역 지정 몸통2 영역 지정 몸통1, 몸통2 영역 결합 몸통1, 몸통2 영역 삭제

[표 3]의 스크립트를 실행하여 얻은 인체 영역 분할 결과는 [그림 18]과 같다.



[그림 18] 인체 영역 분할 결과

2. 영역 분할 활용

위와 같이 분할된 인체 영역을 활용한 예로 허리옆점을 찾는 과정을 스크립트로 작성함으로써 그 유효성을 검증하고자 한다. 먼저, 허리 높이를 찾는 다양한 방법이 있으나 여기에서는 인체 모델에서 정중시상단면을 획득하여 그중 뒷면을 나타내는 곡선에서 가장 오목하게 들어간 점을 찾아 허리뒤점으로 지정하고 허리뒤점을 기준으로 허리앞점, 허리옆점을 찾는 과정으로 진행하였다.

인체 영역 분할 스크립트를 작성했다고 가정한 후, FindCrotch 명령어를 통해 찾은 살점과 오른쪽 방향 벡터 $[1 \ 0 \ 0]$ 을 이용하여 S0라는 정중시상단면을 구한다. 그리고 정중시상단면 S0에서 Y축으로 가장 높은 점을 머리마루점으로 지정한다. 정중시상단면 중 뒷면 곡선만을 획득하기 위해 먼저 정중시상단면을 앞뒷면으로 분류할 관상면을 정의해야 한다. 머리마루점과 살점을 지나는 관상면을 구하기 위해 머리마루점에서 살점으로 향하는 벡터 V1을 구하고, V1 벡터를 X축을 기준으로 90도 회전한 법선 벡터 V2를 구한다. 그리고 머리마루점과 V2 벡터를 이용하여 관상면 PL1를 지정한다. 관상면 PL1를 기준으로 정중시상단면 S0를 앞면 S1과 뒷면 S2로 분리하고, 인체 모델 샘플의 통계로 얻은 허리 높이 범위에서 뒷면 S2에 속하는 점 중 가장 오목한 점을 허리뒤점으로 지정한다.

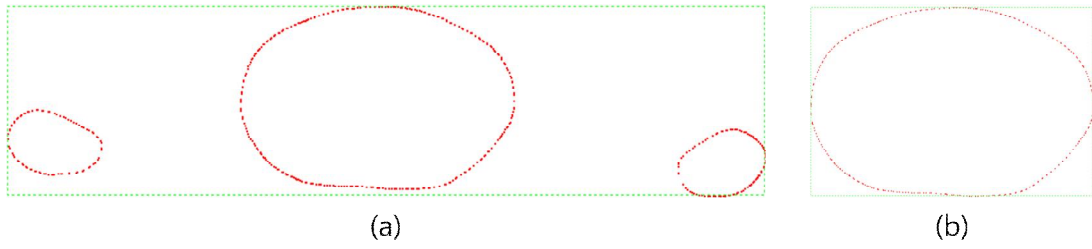
위와 같은 방식으로 찾은 허리뒤점을 지나고 위 방향 벡터 $[0 \ 1 \ 0]$ 을 법선 벡터로 하는 수평면과 분할된 인체 영역 중 오직 몸통과 만나는 단면을 허리단면으로 획득한다. 허리단면과 정중시상단면이 앞면에서 교차하는 점을 허리앞점, 허리 단면에서 x값이 가장 작은 점을 오른허리옆

점, x값이 가장 큰 점을 원허리옆점으로 지정한다. 이를 스크립트로 작성하면 [표 4]와 같다.

[표 4] 허리 관련 랜드마크 검색을 위한 스크립트

스크립트	실행 내용
GetSection,S0,Crotch,[1 0 0] GetMaxYPoint,*Vertex,S	정중시상단면 S0 설정 머리마루점 지정
GetDirection,V1,Vertex,Crotch GetNormal,V2,V1,X SetPlane,PL1,Vertex,V2 DivideSection,S0,PL1,S1,S2 SetHeightRange,WaistH,0.578,0.652 GetMaxZPoint,*BackWaist,S2,WaistH	머리마루점→살점 V1 설정 V1의 법선벡터 V2 설정 PL1 관상면 설정 S0를 PL1으로 분리 허리 높이 범위 설정 허리뒤점 지정
GetSectionRegion,Waist,BackWaist, [0 1 0],Chest GetSectionIntersection,*FrontWaist,S0,Waist ,MaxZ GetMinXPoint,*RightWaist,Waist GetMaxXPoint,*LeftWaist,Waist	허리 단면 설정 허리앞점 지정 허리뒤점 지정 허리옆점 지정

만약, 인체 영역을 몸통과 팔, 다리로 분할하지 않으면 허리 단면을 구할 때 [그림 19(a)]와 같이 허리 단면에 오른팔, 왼팔 영역도 함께 포함되어 허리옆점을 구하기가 위해 영역을 구분하는 별도의 명령어를 매번 작성하여야 한다. 그러나 인체 분할 영역을 활용하여 몸통 영역에서 허리 단면을 구하면 [그림 19(b)]와 같은 허리 단면을 획득할 수 있어 간단한 방식으로 허리옆점을 검색할 수 있다.



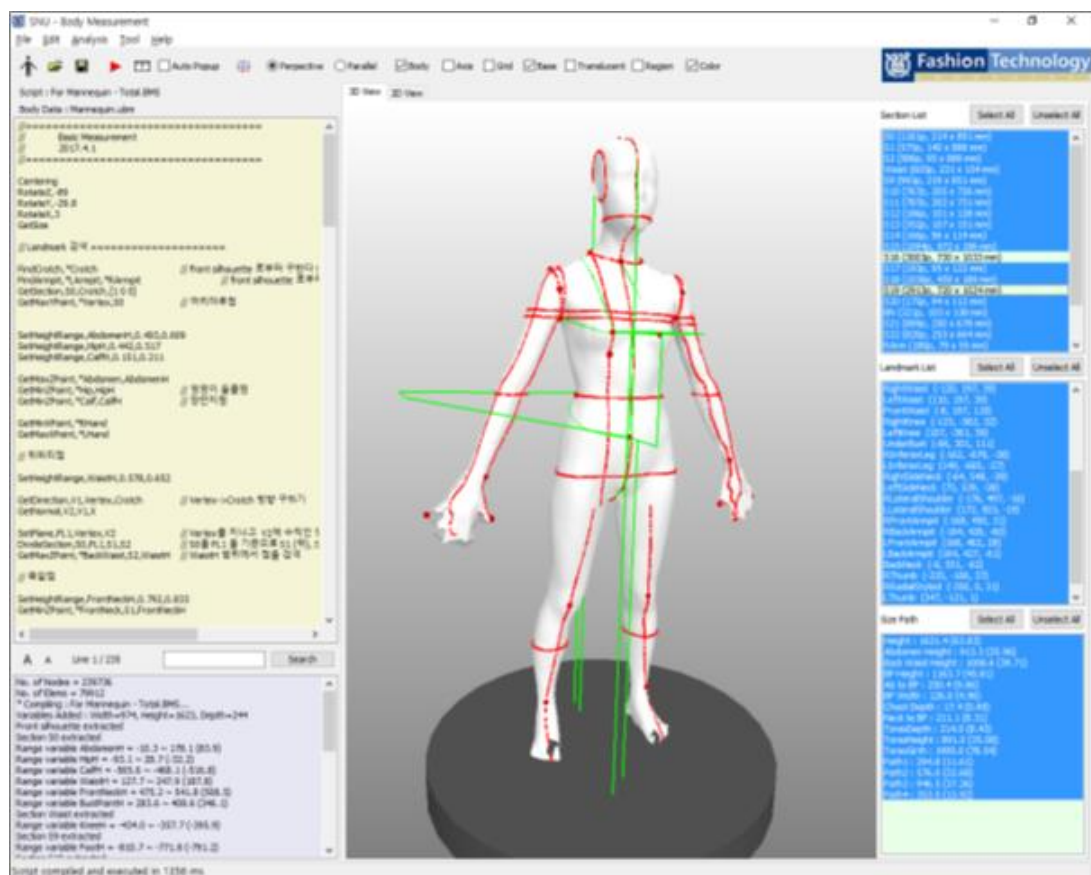
[그림 19] 인체 모델에서 얻은 허리 단면

(a) 인체 분할 영역을 활용하지 않음. (b) 인체 분할 영역을 활용함.

이와 같은 방식으로 분할된 인체 영역을 활용하면 해당 영역 안에서 다양한 랜드마크를 비교적 손쉬운 방법으로 검색할 수 있다.

제 5 절 스크립트 기반의 인체 측정

본 연구에서 개발된 스크립트 기반의 범용 3차원 인체 측정 시스템 개요는 [그림 20]과 같다.

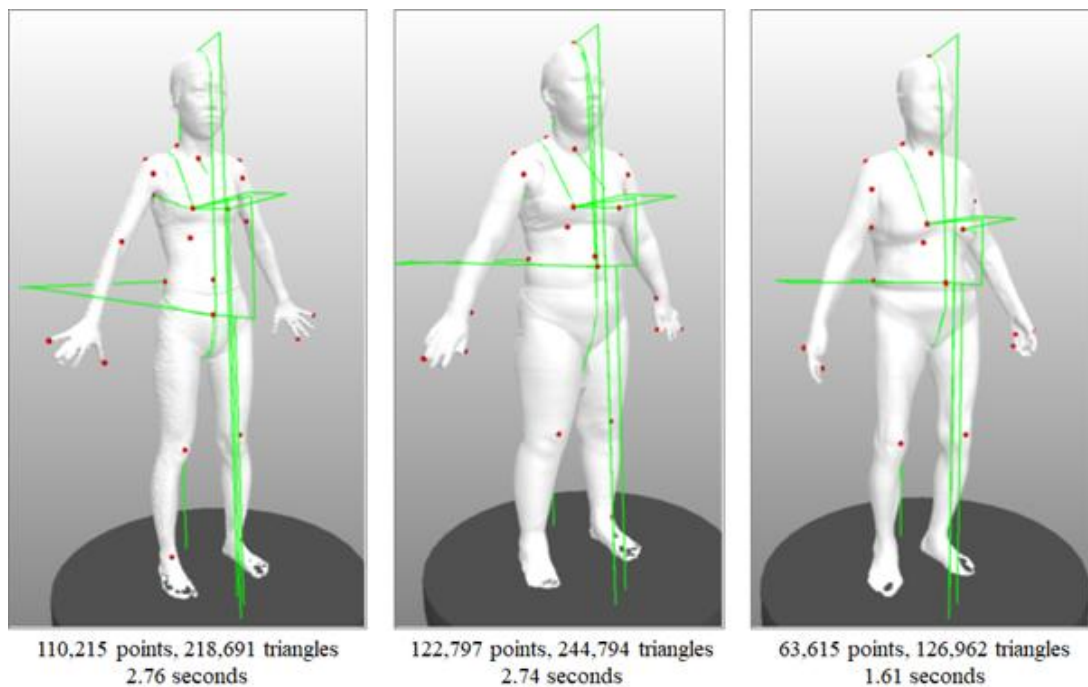


[그림 20] 개발된 3차원 인체 측정 시스템의 개요

본 연구에서 개발한 다양한 명령어를 결합하여 스크립트를 작성한 결과(세부 스크립트 내용은 [부록 2] 참고), 24개의 단면형상을 추출하고,

추출한 단면을 분석하여 34개의 랜드마크를 찾을 수 있었다. 또한, 찾은 랜드마크 사이에 패스를 생성하여, 11개의 간단한 치수와 4개의 복잡한 경로 치수를 측정하였다. 측정에 사용된 마네킹 인체 모델은 239,736개의 포인트와 79,912개의 삼각형 요소로 이루어져 있었으며, 모든 과정은 Intel Core i7-6700 CPU를 사용한 Windows 10 운영체제에서 1.3초라는 비교적 짧은 시간에 완료되었다.

동일한 스크립트를 다른 인체 모델 데이터에 테스트해 본 결과는 [그림 21]과 같다.



[그림 21] 다양한 인체 모델 측정 결과

[그림 21]과 같이 다양한 형태의 모델에 대해서 측정이 잘 이루어지는 것을 볼 수 있다. 또한, 살점과 겨드랑점을 찾는 특수 명령어도 다양한 체형의 데이터에 대해 잘 작동하였다.

본 연구에서는 성인 여자의 체형 측정을 예로 들었지만, 앞 장에서 설명한 바와 같이 다양한 명령어를 결합하여 스크립트를 작성한다면 프로그램을 새로 만들거나 수정하는 복잡한 과정이 없이 개발된 인체 측정 시스템을 활용하여 인종, 연령, 성별 등에 구애받지 않고 연구자의 연구에 필요한 어떤 치수라도 측정할 수 있을 것으로 기대된다.

제 5 장 결론 및 제언

본 연구에서는 인체 스캐너의 종류와 관계없이 랜드마크와 측정치를 자유롭게 정의하고 측정할 수 있는 스크립트 기반의 범용 3차원 인체 측정 시스템을 개발하였다. 이를 위해서 여러 3차원 인체 스캐너에서 얻은 다양한 형식의 스캔 데이터를 사용할 수 있도록 중립적 데이터 구조를 설계하고, 이로 변환하는 인체 모델 데이터 변환 소프트웨어를 개발하였다. 그리고 프로그래밍 언어에 비해 훨씬 간단한 스크립트 언어를 개발하여 프로그래밍에 대한 깊은 지식이 없는 연구자라도 기하학 관련 명령어를 결합하여 자신만의 랜드마크와 측정치를 쉽게 정의할 수 있게 하였다. 또한, 인체 측정에 꼭 필요하지만 정의하기 어려운 살점, 겨드랑점과 같은 랜드마크를 찾는데 특화된 함수들을 개발하였으며, 특히, 직접 만지거나 뼈를 통해 찾을 수 있는 랜드마크를 검색하기 위해 기계 학습으로 얻은 인공신경망 데이터를 활용하여 랜드마크를 검색하는 명령어 역시 개발하였다. 그리고 검색한 랜드마크들 사이에 다양한 패스를 만들어 직선이나 곡선 길이 측정이 가능할 뿐만 아니라 각도 측정, 영역의 부피 계산 등이 가능하도록 개발하였다. 끝으로 사용자가 작업 중에 실시간으로 시각적으로 측정 결과를 확인하여 측정 작업이 용이하도록 개발하였으며, 위에서 언급한 모든 작업이 편리하도록 대화식 사용자 인터페이스를 이용한 통합 개발 환경으로 3차원 인체 측정 소프트웨어를 개발하였다.

본 연구에서 개발된 3차원 인체 측정 시스템을 이용하면 3차원 인체 스캐너의 종류와 관계없이 인체 모델 데이터를 분석할 수 있을 것이며

인종, 연령, 성별 등 다양한 모집단에 맞는 스크립트 데이터베이스를 각각 구축함으로써 인체 측정 연구의 범위를 크게 확장할 수 있을 것으로 기대된다. 향후 추가연구를 통해 인체 표면 형상 분석, 표면적 계산, 체표면 전개를 통한 평면 패턴 획득 등의 3차원 인체 모델 데이터의 동적 측정 및 분석이 가능하도록 그 기능을 확대하고자 한다.

참 고 문 헌

1. N. Werghi, "Segmentation and Modeling of Full Human Body Shape From 3-D Scan Data: A Survey", IEEE Transactions on Systems, Man, And Cybernetics-Part C: Applications and Reviews, vol. 37, no. 6, pp. 1122-1136, 2007.
2. P. Treleaven, J. Wells, "3D Body Scanning and Healthcare Applications", Computer, IEEE Computer Society, vol. 40, no. 7, pp. 28-34, 2007.
3. N. D'Apuzzo, "3D Body Scanning Technology for Fashion and Apparel Industry", In Proceedings of IS&T/SPIE Electronic Imaging 2007, vol. 6491, 2007.
4. A. Petrova, S. P. Ashdown, "Three-Dimensional Body Scan Data Analysis: Body Size and Shape Dependence of Ease Values for Pants' Fit", Clothing and Textiles Research Journal, vol. 26, pp. 227-252, 2008.
5. H. Han, Y. Nam, "Automatic body landmark identification for various body figures", International Journal of Industrial Ergonomics, vol. 41, no. 6, pp. 592-606, 2011.
6. P. Devarajan, C. L. Istook, "Validation of Female Figure Identification Technique (FFIT) for Apparel Software", Journal of Textile and Apparel, Technology and Management, vol. 4 no. 1, pp. 1-22, 2004.
7. K. P. Simmons, C. L. Istook, "Body measurement techniques: Comparing 3D body-scanning and anthropometric methods for

- apparel applications”, *Journal of Fashion Marketing and Management: An International Journal*, vol. 7, no. 3, pp. 306–332, 2003.
8. L. Dekker, I. Douros, B. F. Buxton, P. Treleaven, “Building Symbolic Information for 3D Human Body Modeling from Range Data”, *Proceedings of the Second International Conference on 3-D Digital Imaging and Modeling (3DIM'99)*, pp. 388–397, 1999.
 9. R. Suikerbuik, H. Tangelder, H. Daanen, A. Oudenhuijzen, “Automatic Feature Detection in 3D Human Body Scans”, *SAE 2004 Transactions Journal of Aerospace*, vol. 113, no. 1, pp. 260–263, 2004.
 10. Z. B. Azouz, C. Shu, A. Mantel, “Automatic Locating of Anthropometric Landmarks on 3D Human Models”, *Third International Symposium on 3D Data Processing, Visualization, and Transmission*, pp. 750–757, 2006.
 11. A. Giachetti, E. Mazzi, F. Piscitelli, M. Aono, A.B. Hamza, T. Bonis, P. Claes, A. Godil, C. Li, M. Ovsjanikov, et al., “SHREC'14 Track: Automatic Location of Landmarks used in Manual Anthropometry”, *Proceeding of the 7th Eurographics Workshop on 3D Object Retrieval*, pp. 93–100, 2014.
 12. W. Hargassner, T. Hofer, C. Klammer, J. Pichler, G. Reisinger, “A Script-Based Testbed for Mobile Software Frameworks”, *2008 1st International Conference on Software Testing, Verification, and Validation*, pp. 448–457, 2008.
 13. S. Chaudhury, S. Lyskov, J. J. Gray, “PyRosetta: a

- script-based interface for implementing molecular modeling algorithms using Rosetta”, *Bioinformatics*, vol. 26, no. 5, pp. 689–691, 2010.
14. C. Geuzaine, J. Remacle, “Gmsh: A 3-D finite element mesh generator with built-in pre- and post-processing facilities”, *International Journal for Numerical Methods in Engineering*, vol. 79, no. 11, pp. 1309–1331, 2009.
 15. C. L. Istook, S. Hwang, “3D body scanning systems with application to the apparel industry”, *Journal of Fashion Marketing and Management: An International Journal*, vol. 5 no. 2, pp. 120–132, 2001.

Abstract

Development of a Script based Universal Three-Dimensional Body Measurement System

Minsuk Kim

Dept. of Textiles, Merchandising and Fashion Design

The Graduate School

Seoul National University

A software system has been developed that can analyze three-dimensional (3D) human body scan data regardless of 3D human body scanner type. A neutral 3D model data structure as well as a data conversion software were designed to input and output various types of 3D human body scan data. Users can define custom landmarks and measurement items using a developed script language without having to modify the software. In addition to the general 3D geometry analysis functions, some special functions have also been developed to find some important landmarks such as crotch or axilla points accurately and to find some landmarks using artificial neural networks. Measured results were visualized in many ways to

facilitate the analysis process. 3D body measurement system has been developed as an Integrated Development Environment to make smooth all of these tasks.

keywords : Three-Dimensional Body Scan Data, Anthropometry, Unicursal Software, Script based Measurement, Landmark Detection, Integrated Development Environment

Student Number : 2016-25278

부 록

[부록 1] 명령어 리스트

명령	파라미터	설명
Centering	XYZ or X or YZ etc.	모델의 중심을 좌표축 중심으로 이동
Move	x,y,z (좌표값)	모델을 x,y,z 좌표값만큼 이동
RotateX/Y/Z	각도 (degree)	X/Y/Z축을 기준으로 모델을 회전
Scale	배율	모델을 배율만큼 확대 or 축소
GetSize		모델의 크기를 계산 Width, Height, Depth, Top, Bottom 내부 변수에 저장
GetDistance	변수이름,점1,점2	점1과 점2 사이 거리 계산해서 변수 에 저장
점.x/.y/.z		점 or 랜드마크의 x/y/z 좌표 참조
SetHeightRange	변수이름,최소비율,최대비율	Height 를 기준으로 y 축 범위 (Height*최소비 ~Height*최대비율)를 지정하여 변수에 저장
변수.min/max		범위가 주어진 변수의 최소/최대값 참조
SetRegion	영역이름,중심1,법선벡터1, 중심2,법선벡터2,...	(중심 ,법선벡터)로 정의되는 평면 위쪽 공간들의 교집합으로 영역을 설정
SetSearchRange	영역이름,하한점,상한점 FULL	점 사이 육면체 형태 검색 영역 지정 FULL를 사용하면 전 영역 지정
UnionRegion	영역이름,영역1,영역2,...	여러 영역의 합집합으로 새로운 영역 지정
DeleteRegion	영역1,영역2,...	지정된 영역을 삭제
SetSearchRegion	영역1,영역2,...	검색 범위를 영역으로 한정
SetPoint	점이름,[x y z 점이름,점	x,y,z 좌표의 점 생성, 점이름 앞에 * 를 붙이면 랜드마크가 됨
SetPointDivide	점이름,점1,점2,m,n	두 점 사이를 m:n 으로 내분하는 점 생성

명령	파라미터	설명
GetDirection	점이름,점1,점2	점1에서 점2로 향하는 방향 벡터의 좌표로 점 생성
SetPointInverse	점이름,점	점의 반대 방향 벡터로 점 생성
RotatePoint	점,x각도,y각도,z각도	지정된 점을 X,Y,Z축 방향으로 회전
MovePoint	점,dx,dy,dz	지정된 점을 dx,dy,dz만큼 이동
GetNormal	점이름,점,X/Y/Z	지정된 점을 X/Y/Z 축 기준 반시계 방향으로 90도 회전한 점 생성
GetSection	단면이름,점,법선벡터	점과 법선벡터로 이루어진 평면으로 모델을 자른 단면을 생성
GetSectionRegion	단면이름,점,법선벡터,영역	주어진 영역 안에서 점과 법선벡터로 이루어진 평면으로 모델을 자른 단면을 생성
GetSection3P	점1,점2,점3	3개의 점으로 이루어진 평면으로 모델을 자른 단면을 생성
단면.width/ .height/.girth		단면의 폭/높이/둘레 참조
SetPlane	평면이름,점,법선벡터	점과 법선벡터를 갖는 평면을 생성
DivideSection	단면,평면,단면이름1, 단면이름 2	단면을 평면으로 분할하여 법선벡터 방향의 단면을 단면1에, 그 반대 방향을 단면2에 저장
GetFarthestPoint	점이름,단면,점1,점2	단면 위의 점 중에서 점1 과 점2를 연결하는 선분과 가장 멀리 떨어진 점을 구함
GetFarthestPoint Sil	점이름,Front/Side,점1,점2	Font/Side 실루엣의 주어진 점 구간에서 두 점을 연결하는 선분과 가장 멀리 떨어진 점을 구함
GetSectionInter section	점이름,단면1,단면2,조건 (MaxX/MinY/MaxZ etc.)	단면1과 2의 교점 중에서 조건에 맞는 점을 구함
GetSectionDivide Point	점이름,단면,점1,점2,m,n, 조건(MaxX/MinY etc.)	단면에 점1과 2 사이를 m:n 으로 내분한 뒤 조건에 맞는 점을 구함
GetSectionCenter	점이름,단면이름	지정된 단면의 중심을 점이름으로 저장
GetConvexHull	단면1,단면2,...	단면을 둘러싸는 convex hull를 생성

명령	파라미터	설명
GetMaxPeak2D	점이름,단면,X/Y	평면에 투영된 단면에서 X/Y축 방향 좌표가 최대인 점을 찾음
GetMinPeak2D	점이름,단면,X/Y	평면에 투영된 단면에서 X/Y축 방향 좌표가 최소인 점을 찾음
SortSection2D	단면이름,X/Y	X/Y 좌표를 기준으로 2D단면의 점을 정렬
SetPeakSearch Threshold	수식	Peak를 찾을 때 x/y 좌표값 차이가 수식 이상 떨어진 점은 연속된 점으로 간주하지 않음
SetClosedSearch Threshold	수식	Closed Path를 찾을 때 수식 이상 떨어진 점은 연속된 점으로 간주하지 않음
GetMaxX/Y/Z Point	점이름,단면 점이름,범위변수(Height) 점이름,단면,범위변수	단면 또는 Height Range 안에서 X/Y/Z 좌표 중 가장 큰 값을 가진 점을 저장
GetMinX/Y/Z Point	점이름,단면 점이름,범위변수(Height) 점이름,단면,범위변수	단면 또는 Height Range 안에서 X/Y/Z 좌표 중 가장 작은 값을 가진 점을 저장
SetResult	결과이름,수식	수식값을 Result List에 추가
SetLandmark	랜드마크이름,점	점을 새로운 랜드마크로 정의
SetHeightResult	결과이름,점1 결과이름,점1,점2	바닥면에서 점1까지 높이 점1과 2 사이의 높이
SetWidthResult	결과이름,점1,점2	점1과 2 사이의 폭(X 좌표 차)
SetDepthResult	결과이름,점1,점2	점1과 2 사이의 깊이(Z 좌표 차)
SetDistance Result	결과이름,점1,점2	점1과 2 사이의 직선거리
SetSectionPath	결과이름,단면	단면의 convex hull 을 경로로 표시
SetSection Result3	결과이름,점1,점2,점3, Min/Max	세 점을 지나는 convex hull 중의 짧은/긴 경로를 결과로 저장하고 경로로 표시
SetSection Result2	결과이름,점1,점2, Front/Side/Top,Min/Max	앞/옆/뒷면에서 두 점을 연결하는 단면으로 자르고 짧은/긴 경로를 저장하고 경로 표시

명령	파라미터	설명
SetSubPathResult	결과이름,Path이름, 시작점,끝점	지정된 Path의 시작점에서 끝점까지를 새로운 Path로 만들고 결과에 추가
SubdividePath	Path이름,Segment길이	Path를 작은 길이로 나누어 부드럽게 구함
SetAngleResult	결과이름,점1,점2,점3, Front/Side/Top	점1에서 2로 가는 벡터와 점3이 이루는 각을 Front/Side/Top에서 봤을 때 각도를 구하고 경로를 추가
SetSplinePath Result	결과이름,점1,점2,...,점n	세 개 이상의 점이 필요하며 점1과 n이 같으면 closed path로 간주
FindCrotch	점이름	전면 실루엣에서 Crotch 랜드마크를 구함
FindArmpit	왼쪽겨드랑점이름, 오른쪽겨드랑점이름	전면 실루엣에서 Armpit 랜드마크를 구함
LoadLandmark		인체 데이터 파일과 이름이 같은 txt 파일의 랜드마크 좌표를 읽음.
SaveLandmark		현재 랜드마크의 좌표를 인체 데이터 파일과 같은 이름의 .csv파일로 저장
SaveResult		현재 측정 데이터를 바디파일 이름과 같은 .csv 파일로 저장
LoadTable	테이블이름,파일이름	스크립트 폴더에서 테이블을 읽음.
GetSectionsFrom Table	테이블이름,ID or CURR_BODY,offset	테이블 ID 값으로 섹션 생성 높이에 offset을 더함
SaveRegion AsMSH	영역이름	지정된 영역을 바디파일_영역이름 .msh로 저장
CalculateVolume	파일이름	.msh형태의 메쉬 파일을 통해 영역 의 부피를 구함
FormFFD	[x1 x2 ...],[y1 y2 ...],[z1 z2 ...]	최소/최대값 사이에 생성할 격자의 x y z 좌표 모두 작은 것에서 큰 것 순으로 형성
MoveLattice	[u v w],점이름 or [x y z]	해당 control point를 x y z만큼 이동
UpdateLattice		Lattice에 맞게 물체를 변형
LoadANN	네트워크이름,파일이름	파일이름의 인공신경망 학습 데이터를 읽어서 네트워크를 추가
GetANNPoint	네트워크이름,점이름, 점1,점2,...	지정된 네트워크와 학습에 사용된 여러 점을 입력하면 점이름의 랜드마크 생성

[부록 2] 분석에 사용한 스크립트의 예

```
// Initial Alignment =====
Centering
RotateZ,-89
RotateY,-29.8
RotateX,3
GetSize

// Search Landmark =====
FindCrotch,*Crotch
FindArmpit,*LArmpit,*RArmpit
GetSection,S0,Crotch,[1 0 0]
GetMaxYPoint,*Vertex,S0
SetHeightRange,AbdomenH,0.493,0.609
SetHeightRange,HipH,0.442,0.517
SetHeightRange,CalfH,0.151,0.211
GetMaxZPoint,*Abdomen,AbdomenH
GetMinZPoint,*Hip,HipH
GetMinZPoint,*Calf,CalfH
GetMinXPoint,*RHand
GetMaxXPoint,*LHand

// Posterior Waist
SetHeightRange,WaistH,0.578,0.652
GetDirection,V1,Vertex,Crotch
GetNormal,V2,V1,X
SetPlane,PL1,Vertex,V2
DivideSection,S0,PL1,S1,S2
GetMaxZPoint,*BackWaist,S2,WaistH
```

```

// Anterior Neck
SetHeightRange,FrontNeckH,0.792,0.833
GetMinZPoint,*FrontNeck,S1,FrontNeckH

// Nipple
SetHeightRange,BustPointH,0.674,0.751
SetSearchRange,[50 -1000 -1000],[1000 1000 1000]
GetMaxZPoint,*LeftBP,BustPointH
SetSearchRange,[-1000 -1000 -1000],[-50 1000 1000]
GetMaxZPoint,*RightBP,BustPointH

// Segmentation =====
SetPoint,LeftD,[-1 0 0]
SetPointDivide,RO,RHand,Crotch,1,1
GetDirection,V5,RArmpit,RO
GetNormal,VR,V5,Z
SetPointInverse,VRi,VR
SetPoint,RightD,[1 0 0]
SetPointDivide,LO,LHand,Crotch,1,1
GetDirection,V6,LArmpit,LO
GetNormal,VLi,V6,Z
SetPointInverse,VL,VLi
SetPoint,DownD,[0 -1 0]
SetPoint,UpD,[0 1 0]
SetRegion,RightArm,RArmpit,LeftD,RO,VR
SetRegion,LeftArm,LArmpit,RightD,LO,VL
SetRegion,RightLeg,Crotch,DownD,Crotch,LeftD,RO,RightD
SetRegion,LeftLeg,Crotch,DownD,Crotch,RightD,LO,LeftD
SetRegion,Chest1,Crotch,UpD,RO,VRi,LO,VLi
SetRegion,Chest2,Crotch,UpD,RArmpit,RightD,LArmpit,LeftD

```

```

UnionRegion,Chest,Chest1,Chest2
DeleteRegion,Chest1,Chest2

// Lateral Waist, Anterior Waist
SetSearchRange,Full
GetSectionRegion,Waist,[0 BackWaist.y 0],[0 1 0],Chest
GetMinXPoint,*RightWaist, Waist
GetMaxXPoint,*LeftWaist, Waist
GetSectionIntersection,*FrontWaist,S0,Waist,MaxZ

// Midpatella
SetHeightRange,KneeH,0.232,0.279
SetSearchRegion,RightLeg
GetMaxZPoint,*RightKnee,KneeH
SetSearchRegion,LeftLeg
GetMaxZPoint,*LeftKnee,KneeH
SetSearchRange,Full

// Inferior Breast
GetSectionRegion,S9,RightBP,[1 0 0],Chest
GetSectionIntersection,I0,Waist,S9,MaxZ
GetFarthestPoint,*UnderBust,S9,RightBP,I0

//Inferior Leg
SetHeightRange,FootH,0,0.024
SetSearchRegion,RightLeg
GetMaxZPoint,RBigtoe,FootH
GetMinXPoint,RLittleToe,FootH
SetPointDivide,M0,RBigtoe,RLittleToe,1,2
GetDirection,V7,M0,RightKnee
GetNormal,V8,V7,Z

```



```

GetSection,S10,M0,V8
GetFarthestPoint,*RInferiorLeg,S10,M0,RightKneeSetSearchRegion,LeftLeg
GetMaxZPoint,LBigtoe,FootH
GetMaxXPoint,LLittleToe,FootH
SetPointDivide,M1,LBigtoe,LLittleToe,1,2
GetDirection,V9,M1,LeftKnee
GetNormal,V10,V9,Z
GetSection,S11,M1,V10
GetFarthestPoint,*LInferiorLeg,S11,M1,LeftKnee

// Lateral Neck
SetSearchRange,Full
GetSection,S12,RArmpit,[1 0 0],RightWaist.y,Vertex.y
GetMaxYPoint,RSH,S12
SetPointDivide,IP0,Vertex,FrontNeck,2,1
GetSection,S13,IP0,[0 1 0]
GetMinXPoint,RIP0,S13
GetFarthestPointSil,*RightSideNeck,Front,RSH,RIP0
GetSection,S14,LArmpit,[1 0 0],LeftWaist.y,Vertex.y
GetMaxYPoint,LSH,S14
GetMaxXPoint,LIP0,S13
GetFarthestPointSil,*LeftSideNeck,Front,LIP0,LSH

// Lateral Shoulder
GetSection,S15,RArmpit,[0 1 0]
GetMinXPoint,RIP1,S15
GetDirection,V11,RIP1,RightSideNeck
GetNormal,V11,V11,X
GetSection,S16,RIP1,V11
GetFarthestPoint,RIP2,S16,RIP1,RightSideNeck
SetPointDivide,RLS,RIP2,RSH,1,1

```

GetDirection,V12,RArmpit,RLS
 GetNormal,V12,V12,Z
 GetSection,S17,RArmpit,V12,RightBP.y,Vertex.y
 GetMaxYPoint,*RLateralShoulder,S17
 GetSection,S18,LArmpit,[0 1 0]
 GetMaxXPoint,LIP1,S18
 GetDirection,V13,LIP1,LeftSideNeck
 GetNormal,V13,V13,X
 GetSection,S19,LIP1,[0 V13.y V13.z]
 GetFarthestPoint,LIP2,S19,LIP1,LeftSideNeck
 SetPointDivide,LLS,LIP2,LSH,1,1
 GetDirection,V14,LArmpit,LLS
 GetNormal,V14,V14,Z
 GetSection,S20,LArmpit,V14,LeftBP.y,Vertex.y
 GetMaxYPoint,*LLateralShoulder,S20

 //Anterior Midaxilla, Posterior Midaxilla
 GetSectionDividePoint,*RFrontArmpit,S17,RLateralShoulder,RArmpit,1,1,MaxZ
 GetSectionDividePoint,*RBackArmpit,S17,RLateralShoulder,RArmpit,1,1,MinZ
 GetSectionDividePoint,*LFrontArmpit,S20,LLateralShoulder,LArmpit,1,1,MaxZ
 GetSectionDividePoint,*LBackArmpit,S20,LLateralShoulder,LArmpit,1,1,MinZ

 // Cervicale
 SetHeightRange,BackNeckH,0.825,0.862
 SetSearchRegion,Chest
 GetSection3P,BN,FrontNeck,RightSideNeck,LeftSideNeck
 GetSectionIntersection,*BackNeck,BN,S0,MinZ

 // Radial Styloid
 SetHeightRange,HandH,0.338,0.45
 SetSearchRegion,RightArm

```

GetMaxXPoint,*RThumb,HandH
GetDirection,V15,RThumb,RArmpit
GetNormal,V15,V15,X
GetSection,S21,RThumb,V15
SetClosedPathThreshold,200
GetFarthestPoint,*RRadialStyloid,S21,RThumb,RArmpit
SetClosedPathThreshold,10000
SetSearchRegion,LeftArm
GetMinXPoint,*LThumb,HandH
GetDirection,V16,LArmpit,LThumb
GetNormal,V16,V16,X
GetSection,S22,LThumb,V16
SetClosedPathThreshold,200
GetFarthestPoint,*LRadialStyloid,S22,LArmpit,LThumb
SetClosedPathThreshold,10000
GetSectionRegion,RArm,[0 BackWaist.y 0],[0 1 0],RightArm
GetSectionRegion,LArm,[0 BackWaist.y 0],[0 1 0],LeftArm
GetSectionRegion,RLeg,[0 Calf.y 0],[0 1 0],RightLeg
GetSectionRegion,LLeg,[0 Calf.y 0],[0 1 0],LeftLeg
GetSectionRegion,Hip,[0 Hip.y 0],[0 1 0],Chest
GetConvexHull,Hip

// Show results =====
SetHeightResult,Height,Vertex
SetHeightResult,Abdomen Height,Abdomen
SetHeightResult,Back Waist Height,BackWaist
SetHeightResult,BP Height,RightBP
SetHeightResult,Ab to BP,RightBP,Abdomen
SetWidthResult,BP Width,LeftBP,RightBP
SetDepthResult,Chest Depth,Abdomen,BackWaist
SetDistanceResult,Neck to BP,FrontNeck,LeftBP

```

```

SetResult,TorsoDepth,S0.width
SetResult,TorsoHeight,S0.height
SetResult,TorsoGirth,S0.girth
SetSectionPath,TorsoGirth,S0

// Surface measurement =====
SetSearchRegion,Chest
SetPathResult3,Path1,RArmpit,RightBP,LeftBP
SetPathResult3,Path2,BackWaist,Crotch,Abdomen
SetPathResult2,Path3,RightBP,Crotch,Front,Max
SetPathResult2,Path4,BackWaist,Abdomen,Side

```